ARI Research Note 87-84

DTIC FILE COPY

AD-A190 788

HUMAN PROBLEM SOLVING IN COMPLEX DYNAMIC ENVIRONMENTS

William B. Rouse and Richard L. Henneman
Georgia Institute of Technology

DTIC ELECTE FEB 10 1988 S D

for

Contracting Officer's Representative
Michael Drillings

BASIC RESEARCH LABORATORY
Michael Kaplan, Director

U. S. Army

Research Institute for the Behavioral and Social Sciences

January 1988

Approved for public release; distribution unlimited.

88 2 08 041

U. S. ARMY RESEARCH INSTITUTE
FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency under the Jurisdiction of the
Deputy Chief of Staff for Personnel

EDGAR M. JOHNSON
Technical Director

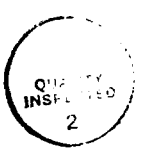
WM. DARRYL HENDERSON
COL, IN
Commanding

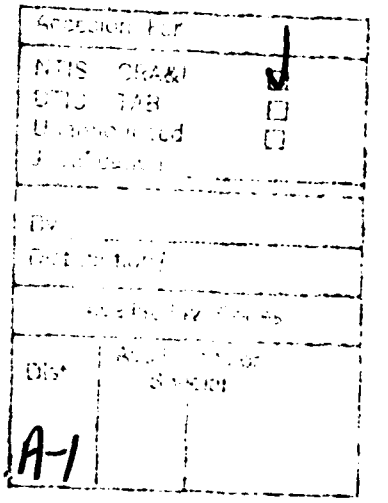
Research accomplished under contract
for the Department of the Army

Georgia Institute of Technology

Technical review by

Dan Ragland





This report, as submitted by the contractor, has been cleared for release to Defense Technical Information Center (DTIC) to comply with regulatory requirements. It has been given no primary distribution other than to DTIC and will be available only through DTIC or other reference services such as the National Technical Information Service (NTIS). The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other official documentation.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ARI Research Note 87-84	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Human Problem Solving in Complex Dynamic Environments		5. TYPE OF REPORT & PERIOD COVERED Interim Report June 84 - May 85 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) William B. Rouse and Richard L. Henneman		8. CONTRACT OR GRANT NUMBER(s) MDA903-82-C-0145
9. PERFORMING ORGANIZATION NAME AND ADDRESS Center for Man-Machine Systems Research Georgia Institute of Technology Atlanta, GA 30332		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 2Q161102B74F
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Research Institute for the Behavioral and Social Sciences, 5001 Eisenhower Avenue, Alexandria, VA 22333-5600		12. REPORT DATE December 1987 13. NUMBER OF PAGES 76
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) - -		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE - -

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

- -

18. SUPPLEMENTARY NOTES

Michael Drillings, contracting officer's representative

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Human Performance
Problem Solving
Man-Machine
Computer Models

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This research note summarizes three years of a four year contract to study ways of improving human performance in highly integrated systems in such areas as communications, transportation, manufacturing, etc. Rule-based computer models of human performance (CAIN) are discussed, as are methods for measuring the complexity of the task of monitoring these large-scale systems. Finally, the development of a computer model (MABEL) which requires subjects to monitor a large-scale communications network is described.

Keywords: man machine

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

INTRODUCTION

Current trends in computer and communications technology are leading to the development of many highly integrated systems in the domains of communications, transportation, manufacturing, etc. Most of these systems can be represented as large networks of nodes and arcs where nodes denote people, destinations, or machines and arcs denote communication lines, transportation routes, or a variety of activities. Because these systems are highly integrated, it is not unusual for there to be hundreds or thousands of nodes and arcs. Networks of this size and level of connectivity are very complex systems.

Complexity is further increased by the dynamic nature of these networks. The states of the nodes and arcs (i.e., levels, flows, etc.) usually evolve in time and are not amenable to instantaneous control. Further, the demands placed upon the networks are often time-varying, with occurrences of peak demands not always being predictable.

This program of research is concerned with the problem solving behavior of the human whose role is network controller or operator. The job of the network controller is to manage the assets of the network (i.e., nodes and arcs) so as to maximize network efficiency. Further, during peak demand periods, the controller may have to implement control procedures such as load shedding and priority scheduling to assure that overloads do not degrade network performance.

For many aspects of this job, the network controller has computer aids or, in fact, may simply have to monitor an automated system which performs many of the above functions. However, system failures or unusual environmental demands can require that the human intervene and manually control the network. The human's abilities to solve these types

of problem are not well understood. In fact, human problem solving in complex dynamic environments is an area where few research results are available. This area is the topic of the research program whose progress is reported here.

PROGRESS

This section briefly summarizes progress during the first three years of this four-year program of research. Considerably more detail about the most recent results can be found in the papers included in the Appendix.

Most of the first year was devoted to developing an experimental scenario and evaluating the impact of its parameters on human problem solving performance [Henneman and Rouse, 1984a]. Communications networks were chosen as the experimental context. After reviewing a variety of documentation on human control tasks in both commercial and military communications networks, an experimental scenario called MABEL was designed and programmed. MABEL requires subjects to monitor a large-scale automated communications network via a hierarchical multi-page CRT display. Much as discussed in the Introduction, subjects have to manage network assets and, in the event of a failure, intervene to diagnose the failure, compensate for its impact, and restore normal operation.

For the first formal experiment with MABEL, the effects of three independent variables were studied: 1) number of nodes per display, 2) number of levels in the display hierarchy, and 3) failure rate per node. Twelve subjects each participated in six experimental sessions. Overall, this initial experiment with MABEL produced two results of particular interest. First, the effects of number of levels in the hierarchy were

often very strong, producing up to a five-fold degradation of performance for a modest change from two to three levels. The second result of note is that rather different strategies seemed best for different combinations of independent variables. This leads to the question of whether humans can be trained to adapt appropriately or if some form of aided adaptation is needed.

The second year of this research involved two efforts. One effort concerned the development of a rule-based model of human problem solving in the MABEL environment [Viteri 1984]. One general impression that emerged from the experiment and the modeling efforts was that MABEL lacked the contextual richness necessary to provide the type of problem solving environment required for this research. Perhaps the best indication of this is the simplicity of Viteri's model even though it compares fairly well with subjects' behavior.

This observation led to a decision to enhance substantially the contextual aspects of MABEL. The second formal experiment [Henneman and Rouse 1984b, 1985, Henneman 1985] used a contextually augmented version of MABEL called CAIN (Contextually Augmented Integrated Network). The scenario contained cues and associative links (e.g., non-varying geographic node names, recurring failures, and non-uniform loading) to produce a higher fidelity simulation. Cluster size was kept constant at 16 so that subjects could learn and recall context-dependent aspects of the system. Experimental variables were number of connections between nodes (high,low) and number of levels (2,3). Eight subjects each participated in thirteen experimental sessions. Results supported those from Experiment One: increasing number of levels degraded performance, as did decreasing the connectivity between nodes.

Efforts in the third year of this research have been directed towards realizing a major objective of the second experiment, namely, to investigate the nature of complexity in a large scale system [Henneman and Rouse 1985, Henneman 1985]. Two dimensions (and associated measures) of complexity were proposed: complexity due to the structure of the system and complexity due to the strategy of the person trying to control the system. Complexity was considered to be a dynamic property of a human-machine system. Complexity is time-dependent and multi-dimensional; thus, time series analysis was used to develop transfer functions relating the two complexity measures to average time to failure diagnosis. Results indicated that the distinction between structural complexity and strategic complexity is appropriate.

Results also emphasized the different implications that complexity may have for normal system operation and human failure diagnosis performance. A very complex system may function quite well under normal operating conditions. The system is able to absorb the effects of failures to a certain extent while maintaining an adequate level of performance. However, when the problem becomes so critical that the human monitor must intervene and find the problem, the task of failure diagnosis may be very difficult. In summary, although certain system design characteristics may help to avoid the short term effects of failures, these same characteristics may have the dual effect of making the human supervisory controller's task more difficult. These results are presented in the paper included in the Appendix [Henneman and Rouse 1985].

Other efforts in the third year of this research have been directed towards the conceptual development of a sophisticated model-based

performance aid for humans monitoring and controlling CAIN. The proposed rule-based model is described in a paper in the Appendix [Henneman 1985b]. The model is characterized by three stages of problem solving (recognition/classification, planning, and execution) that are prioritized according to the model's knowledge about the task and about the system (e.g., contextual relationships among components).

FUTURE WORK

On-line implementation of the model proposed in the paper in the Appendix has just started. Future plans include the experimental evaluation of the model as an on-line performance aid. The proposed experiment should compare the task performance of two groups of subjects, one of which performs without the aid and the other with the aid. An interesting side issue to explore involves the representation and use of the contextual information included in the experimental scenario.

REFERENCES

1. Henneman, R.L. and Rouse, W.B. "Human problem solving in large-scale networks," Proceedings of the 1982 International Large-Scale Systems Symposium, Virginia Beach, October 1982, pp. 293-297.

2. Henneman, R.L. and Rouse, W.B. "Human performance in monitoring and controlling hierarchical large-scale systems," Proceedings of the 27th Annual Meeting of the Human Factors Society, Norfolk, Virginia, October 1983, pp. 685-689.

3. Henneman, R.L. and Rouse, W.B. "Human performance in monitoring and controlling hierarchical large-scale systems," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-14, No. 2, pp. 184-191, March/April 1984a.

4. Henneman, R.L. and Rouse, W.B. "Assessing the complexity of a large-scale system: measures of system structure and human strategy," Proceedings of the 1984 IEEE International Conference on Systems, Man, and Cybernetics, Halifax, Nova Scotia, October 1984b.

5. Viteri, E. "A rule-based model of a human operator in a complex communication network," MSIE Thesis, Georgia Institute of Technology, 1984.

6. Henneman, R.L. " Human problem solving in complex, hierarchical large-scale systems," Ph.D. Thesis, Georgia Institute of Technology, 1985a.

7. Henneman, R.L. "A model of human performance in a large scale dynamic system," Proceedings of the 1985 IEEE International Conference on Systems, Man, and Cybernetics, Tucson, AZ, November 1985b.

8. Henneman, R.L. and Rouse, W.B. "On measuring the complexity of monitoring and controlling large scale systems," submitted for publication, 1985.

APPENDIX

A MODEL OF HUMAN PERFORMANCE IN A LARGE SCALE DYNAMIC SYSTEM

Richard L. Henneman

Center for Man-Machine Systems Research
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30322

Abstract

A model of human performance in monitoring and controlling complex engineering systems is considered from the perspective of implementing the model as an on-line performance aid. Results from the literature are discussed in the context of CAIN, a simulated large scale system that has been used to study human supervisory control performance [1,2]. A rule-based model of human performance in CAIN is proposed and a methodology for evaluation is suggested.

Introduction

Recent trends in automation have facilitated the creation of large, complex engineering systems. Due to the capabilities of computer technology to control a large number of interconnected components, the normal operation of these systems is typically left to an automatic controller. During unforeseen events (such as system failures) that cannot be handled by the computer, however, a human controller must take corrective action. As many have noted [3], this increase in automation is fundamentally changing ways in which people interact with large systems. The human operator no longer is in charge of the routine, continuous control of the system. Rather, the operator is mostly concerned with the unexpected, the unusual, and the non-routine aspects of system control. Requisite human skills for system control are shifting from psychomotor to problem solving [4].

Recent research activity has focused on human supervisory control and problem solving in complex engineering environments [3], although a majority of this work has been restricted to the process control domain. Henneman and Rouse [1,2] discuss human performance in monitoring and controlling a large scale dynamic network (such as a communication network). These systems can be represented as discrete queueing networks. Henneman and Rouse [1,2] have conducted a series of experiments that empirically assessed the effects of system features (such as number of levels and display size) on human performance via two simulated large scale systems called MABEL and CAIN. Other efforts have been directed towards developing and evaluating measures of large scale system complexity. Results to this point have led to a good empirical understanding of the relationship between the structure of the system and human performance.

Given this understanding of human performance in this task, it is now possible to postulate a model of human problem solving in the CAIN environment. Thoughts presented in this paper are directed towards the development of such a model. First, some previous related modeling efforts are reviewed. Second the CAIN environment is briefly described. Finally, the problem solving model is presented and discussed

Background

The model developed in this paper is an extension of a conceptual model of human problem solving proposed by Rouse [5]. Rouse has suggested that problem solving takes place on three levels: 1) recognition and classification, 2) planning, and 3) execution and monitoring Thus, when a problem situation develops, the first task is to detect that the problem exists and to categorize it (recognition and classification). An approach or plan to solving the problem must then be developed (planning), and finally, the plan must be implemented (execution and monitoring). The model is further characterized by its ability to make either a state- or a structure-oriented response, depending on both the system state and the human's level of expertise. The model assumes that humans have a preference for pattern-recognition solutions to problems -- that is, humans prefer to make context-specific state oriented responses to situations. Moreover, the model operates heterarchically at all three problem solving levels almost simultaneously, with situations constantly being re-evaluated relative to their state- or structure-oriented status.

The model as presented by Rouse [5] is explicitly a conceptual realization/ combination of other more restricted problem solving models. Knaeuper and Rouse [6] attempted to implement an operational model of this conceptual framework in a computer simulated process plant environment called PLANT [7]. They developed a rule-based model called KARL (Knowledgeable Application of Rule-based Logic) that controlled the computer simulated process plant. KARL consists of a set of production rules that comprise the knowledge base and a control structure that accesses that knowledge base. KARL's structure is

defined by the three levels of problem solving described above and also four major tasks that are associated with human performance in a process control environment (i.e., transition, steady-state tuning, failure detection and diagnosis, and failure compensation). Thus, changes were made to the originally proposed model in order to accomodate specific characteristics of process plants. In addition, the model does not explicitly incorporate a mechanism to distinguish between state- and structure-oriented responses.

KARL's performance in controlling PLANT was compared with that of human subjects. Overall, the comparison was favorable in terms of such performance measures as plant output and plant stability. An action-by-action comparison between KARL and subjects revealed, however, two major systematic differences: first, subjects tended to be more conservative in terms of selecting levels of system input and output, and second, KARL tended to adjust input and output more frequently than subjects. These findings were probably a result of differences between subjects' underlying performance goals and KARL's goals. KARL possessed mechanisms that always tried to maximize plant production, a strategy which it, unlike subjects, pursued inflexibly. Consequently, KARL tended to be more extreme in terms of accurately following procedures.

Knaeuper and Morris [8] attempted to use KARL as an on-line aid to subjects controlling PLANT. In light of the difficulties PLANT subjects had in accurately assessing situations and following appropriate procedures [9] and since KARL was good at these activities, the use of KARL as an on-line aid was a logical extension. KARL provided three types of aid: 1) situation assessment (i.e., identification of the appropriate procedure), 2) guidance in following procedures, and 3) performance feedback. Comparing performance of subjects who received help from KARL to those who performed unaided, the aided subjects maintained a higher level of plant stability, scored higher on a paper-and-pencil test of system knowledge, and were more successful in diagnosing an unfamiliar system failure.

As Knaeuper and Morris [8] indicate, the interpretation of the results is not straightforward. In fact, they conclude that although this experiment successfully demonstrated the viability of the use of a model-based performance aid, issues related to on-line training and aiding are far from resolved. The framework outlined in this paper is an attempt to further investigate the use of a model-based performance aid in a different task domain. The next section, therefore, describes a simulated large scale system used to study human failure diagnosis performance.

CAIN: A Simulated Large Scale System

Two previous experiments [1,2] have considered human performance in the monitoring and control of a computer simulated large scale system. In the first experiment, subjects supervised an essentially context-free representation of a large scale network called MABEL (Monitoring, Accessing, Browsing, and Evaluating Limits), trying to optimize such system parameters as number of customers served and customer processing time while trying to diagnose system failures. In the second experiment, the MABEL scenario was substantially augmented to produce a higher fidelity system. This new scenario is called CAIN (Contextually Augmented Integrated Network). The remainder of this section provides a brief overview of CAIN. The reader is referred to Henneman and Rouse [2] for more detail.

Overview of CAIN

CAIN is programmed in Pascal on a VAX 11/780 computer and operates in real time. It is structured as a large hierarchical network that can range in size from hundreds to thousands of nodes. Customers travel through the system from a randomly selected source node to a random destination. Subjects monitor this system activity via a CRT display. When they detect a problem in the system (possibly due to a failure), subjects issue an appropriate command through a keyboard to correct and compensate for the abnormal situation. The overall objectives of the operator are: 1) to maximize the number of customers served, and 2) to minimize customer sojourn time.

Because of the network size, it is not possible to display information about all nodes at one time. Thus, nodes are grouped into relatively small networks called clusters. Human operators are restricted to viewing only one cluster at a time on the CAIN display. Clusters are grouped into hierarchic levels.

Effects of node failures

Under normal circumstances, CAIN operates automatically without human intervention. Since the system cannot automatically diagnose and repair failures, the human must monitor the system looking for evidence of failed components. Node failures can occur in two ways. The first is a randomly occurring failure mode caused by malfunctioning equipment. The second type, capacity failure, can be caused by the randomly occurring failures. Each node has a maximum number of customers that it can store at a time. If this limit is exceeded, the node fails. Thus, if a node fails randomly and a customer needs to visit that node, it will be retained at its previous node. This retention will cause the previous node to stop processing customers, which can lead to a capacity failure. In this way, if the operator does not locate failures quickly, the problems will propagate through the system.

Addition of context

Although the physical hierarchical structure of MABEL was preserved, the addition of contextual information to CAIN required changing both interface and system characteristics. In CAIN, for example, each node in the system is identified by a

specific geographic location (for example, nodes in the highest level of the system are labelled Seattle, Chicago, Miami, etc.) In addition, the contextual fidelity was enhanced through the addition of associative links (i.e., memory aids) and cues (i.e., clues to the location of system problems). Associative links were formed by requiring subjects to reference nodes via their geographic label Cues were formed by the introduction of context-dependent events, such as recurring failures and non-uniform loading

A Model of Human Performance in CAIN

Building from the work of Rouse, Knaeuper and Morris [5,6,8], a model is proposed in this section with the intent of supporting human performance in monitoring and controlling CAIN. First, some overall requirements of the model are specified. Second, a specific model is proposed, and finally, the way in which the model can be used as a performance aid will be discussed.

Overall requirements

Before the model is proposed, several requirements that the model should meet are specified in this section. For example, in order to function as a performance aid, the model must be able to represent several different performance strategies. A result from Henneman and Rouse [2] indicated that subjects discovered failures in CAIN using three modes of failure diagnosis: symptomatic, topographic, and serendipitous. Subjects using a topographic strategy trace failure symptoms from higher system levels to lower level causes. Subjects using a symptomatic strategy make a direct mapping from their system structure knowledge to the failed component. A symptomatic diagnosis relies, therefore, on the subject's contextual knowledge of the system. Finally, subjects may also identify failures accidentally or serendipitously. When using this diagnosis mode, subjects locate failures while browsing through the system or while tracing the cause of a different failure.

These failure diagnosis modes are dependent upon an individual subject's understanding of how the system operates as well as the subject's knowledge of the contextual relationship between system components. Therefore, the model should incorporate an explicit representation of both contextual knowledge and task knowledge. Moreover, the model should allow this contextual knowledge to be augmented over time as subjects gain performance expertise.

Finally, the model should represent the way in which subjects prioritize sub-tasks in monitoring and controlling the network. Multiple system failures and the dynamic nature of the system may cause the operator to have several sub-tasks to perform at any one time. At one instant, for example, the system may have multiple failure symptoms on the display, heavy customer demands in one part of the system, and a failed node. The relative importance of each of these sub-tasks can vary with time. In some cases there will be clear-cut choices among alternatives while in other cases there will be indifference. It may be argued that the essence of good performance in this task is the subject's ability to prioritize sub-tasks that are present concurrently. An important goal of this model, therefore, is to represent sub-task prioritization.

In summary, underlying the development of this model is the demonstration of how a general representation of human problem solving [5] can be adapted to model human performance in a complex large scale environment. More importantly, this model is to be used as an on-line performance aid. Unlike the work of Knaeuper and Morris [8] in which a performance model was adapted post hoc to serve as an aid, the development of this model is motivated by the desire to use it as an on-line aid. In order to achieve these underlying goals, the model should flexibly support several performance strategies, explicitly represent contextual knowledge, and contain a mechanism to prioritize sub-tasks.

A model

A model that meets these requirements is shown in Figure 1. As in Knaeuper and Rouse's KARL [6], the model proposed here will be represented as a set of if-then rules organized into a hierarchical structure. The model contains two levels of activities. The lowest level of the model consists of the three stages of problem solving discussed by Rouse [5]: recognition/classification, planning, and execution. Because multiple sub-tasks may concurrently exist, the model can be operating in any of these stages. Thus, the highest level of the model contains a mechanism to prioritize the performance of sub-tasks in the three lower level stages. The remaining model component represents the

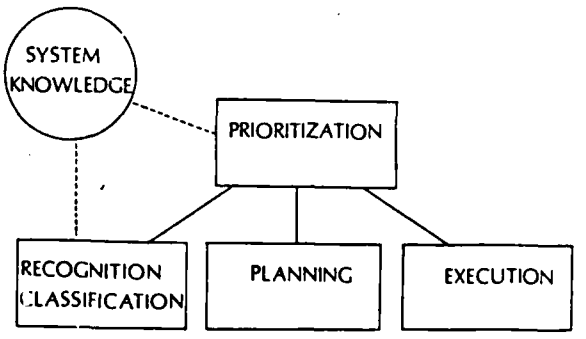


Figure 1. A model of human performance in CAIN

system (or contextual) knowledge needed to perform the task The following paragraphs will discuss each part of the model in more detail.

Recognition/classification takes place when the subject identifies that an event has occurred or a situation exists. Examples include node failure, failure symptoms, abnormal customer demands, and normal situations.

Once an event has been recognized and classified, the subject must develop an approacn to improve the situation. A difference exists between planning at this level and the prioritization or the coordination of low level plans that occurs at the highest model level. Plans at this low level are best compared to simple scripts [10] or short sequences of actions. To illustrate, consider a situation in which a subject observes an increasing queue size in a node. This event suggests that a failure exists in a lower system level; a suitable plan of action would be to 1) display the lower level cluster, and 2) test the new cluster for failed components.

After a suitable course of action is identified, the plan must be implemented. An analysis of the timing of subject's commands [11] indicated that these command sequences are frequently issued in rapid succession, suggesting that the plans are executed automatically with little conscious attention. In the execution phase of this model these command sequences are issued. The assumption is made that once the sequence is started, it cannot be interrupted.

These three phases - recognition/ classification, planning and execution - form the basis of activities implemented by the model. In general, the performance of each sub-task will progress sequentially through each of the three stages. Nevertheless, since multiple sub-tasks may exist, it is likely that this sequential process may be interrupted by a new sub-task of greater importance. As mentioned previously, the key to good performance in this task is the ability to prioritize sub-tasks. Thus, perhaps the most important feature of this model is the way in which activities are prioritized. Prioritization takes place in the highest model level.

Two other components are included in the model, one of which is implicitly embedded within other model components, the other of which is explicitly represented. These two parts represent the knowledge necessary to perform the task: task knowledge and system knowledge.

Task knowledge (analogous to Anderson's procedural knowledge [12]) encompasses the knowledge of how to do things, for example, how to diagnose a failed component. This knowledge will be embedded in the productions (or if-then rules) associated with the model components (i.e., prioritization, recognition/classification, and planning).

System knowledge (analogous to Anderson's declarative knowledge [12]) encompasses the knowledge of contextual relationships among system components. For example, system knowledge might contain a fact like "Evanston is a second level city that is associated with Chicago." In addition to this static knowledge of system structure, system knowledge also encompasses facts that are related to the system dynamics. For example, the names of nodes with recurring problems or regions with high customer loading are likely to be remembered by subjects. In this model, these facts will be stored as system knowledge. This type of knowledge should only be accessed by the recognition/classification and prioritization model components. Planning and execution are performed independently of contextual knowledge. Methods of representing this task system knowledge are currently being investigated [13], along with ways in which the system knowledge can be augmented as subjects gain more expertise.

The model as an aid

This section considers each of the model components and the roles they could play in providing performance assistance. Recognition/classification, for example. will indicate "trouble spots". i.e., regions with a greater liklihood of having a failure or heavy loading. When scanning a display, a subject can easily miss a salient cue. The model should be helpful in terms of indicating those cues that have the greatest likelihood of reflecting a failure.

The planning module can assist by telling operators what to do once they have recognized a situation. This information would be most useful for novice operators. Nevertheless, for some situations that are seen only infrequently, this advice would be useful for all operators. This lack of emphasis on procedural information is markedly different from the advice given by KARL to PLANT subjects [8]. A large part of the advice that KARL provided was procedural information.

Probably the most important aid that the model can offer is in prioritizing sub-tasks. Certain situations are more critical than others; the model should be useful in identifying those sub-tasks that are most important.

Other ways in which the aid should be used is by giving performance feedback and contextual information. As in the KARL-PLANT experiment [8], subjects should receive feedback relative to the success of their actions. In addition, since operators' system knowledge is inevitably at various levels of completeness, the model should assist in augmenting the deficiencies.

The biggest problem in using a model like the one proposed in this paper as an aid is the development of an effective interface between the model and the operator. Since the system state is constantly changing, the information provided by the aid will also be constantly changing. Advice that is relevant

at one time may be erroneous after a few moments. At this time, it is unclear how to control the display of this information. Moreover, since the current CAIN display is already very crowded with verbal information, the addition of advice from a model-based aid may only serve to degrade performance by overloading the human's information processing capbilities. Perhaps synthesized voice output would be a suitable means of presenting this advice.

Future work

On-line implementation of the model proposed in this paper has just started. Future plans include the experimental evaluation of the model as an on-line performance aid. The proposed experiment should compare the task performance of two groups of subjects, one of which performs without the aid and the other with the aid. If the aid proves to be successful, another experiment will evaluate the usefulness of individual model components of the aiding scheme.

Acknowledgement

This work was supported by the Army Research Institute for the Behavioral and Social Sciences under Contract MDA 903-82-C-0145.

References

(1) R.L. Henneman and W.B. Rouse, "Human performance in monitoring and controlling hierarchical large-scale systems." IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-14, no. 2, pp. 184-191, March/April 1984.

(2) R.L. Henneman and W.B. Rouse, "On measuring the complexity of monitoring and controlling large scale systems," submitted for publication.

(3) J. Rasmussen and W.B. Rouse, eds., Human Detection and Diagnosis of System Failures. New York: Plenum Press, 1981.

(4) C D Wickens, Engineering Psychology and Human Performance, Columbus: Charles E. Merrill, 1984.

(5) W.B. Rouse, "Models of human problem solving: detection, diagnosis, and compensation for system failures," Automatica, vol. 19, no. 6, pp. 613-625, November 1982.

(6) A. Knaeuper and W B. Rouse, "A rule-based model of human problem solving behavior in dynamic environments." IEEE Transactions on Systems, Man, and Cybernetics, vol. 15, no. 6, November/December 1985.

(7) N.M. Morris, W.B. Rouse and J.L. Fath, "PLANT: an experimental task for the study of human problem solving in process control," IEEE Transactions on Systems, Man, and Cybernetics, vol. 15, no. 6, November/December 1985.

(8) A. Knaeuper and N.M. Morris, "A model-based approach for online aiding and training in process control," in Proceedings of the 1984 IEEE International Conference on Systems, Man, and Cybernetics, Halifax, NS, pp 173-177, 1984

(9) N.M. Morris and W.B. Rouse, "The effect of type of knowledge on human problem solving in a process control task," IEEE Transactions on Systems, Man, and Cybernetics, vol. 15, no 6, November/December 1985

(10) R.C Schank and R.P. Abelson, Scripts Plans, Goals, and Understanding, Hillsdale, NJ: Lawrence Erlbaum, 1977.

(11) R.L. Henneman, "Human problem solving in complex hierarchical large scale systems," Ph.D. thesis, Georgia Institute of Technology, January 1985.

(12) J.R. Anderson, Language, Memory, and Thought, Hillsdale, NJ: Lawrence Erlbaum, 1976.

(13) R.C. Andes, "Effects of contextual knowledge on human problem solving in large scale systems," M.S. thesis in progress, Georgia Institute of Technology.

ON MEASURING THE COMPLEXITY OF MONITORING AND CONTROLLING LARGE SCALE SYSTEMS

Richard L. Henneman

William B. Rouse

Center for Man-Machine Systems Research

School of Industrial and Systems Engineering

Georgia Institute of Technology

Atlanta, GA 30332

ABSTRACT

The complexity of monitoring and controlling a large scale system, such as a communication network, is considered. Relevant literature is reviewed, with emphasis on both behavioral and non-behavioral approaches to measuring complexity. A simulated large scale network is described that is used in an experiment to assess the effect of network redundancy and number of system levels on human fault diagnosis performance. Experimental data is also used to evaluate two time-varying measures of task complexity (using ANOVA and time-series analysis). The first measure is dependent upon the structure of the system; the second measure is dependent on the strategy of the person controlling the system. Results suggest that this distinction is appropriate. In addition, results emphasize the different implications that complexity can have for normal system operation and human failure diagnosis performance. Although system design characteristics such as redundancy may help to avoid the short term effects of failures, these same characteristics may have the dual effect of making the human supervisory controller's task more difficult.

INTRODUCTION

Recent trends toward increased automation in large scale engineering systems are causing a parallel shift in the role that humans play in these systems. People are increasingly being required to interact with systems only during unforeseen events, such as when a part of the system fails. During these times, proper system functioning is dependent upon the human's decision making and problem solving skills. These human abilities can be enhanced or degraded by a parallel shift in display capabilities: not only is the computer changing the level of automation in systems, but it is fundamentally changing the nature of communication between the human and the system. These changes have the potential of producing tasks of possibly enormous complexity. In light of this potential, it is of basic importance to consider human abilities in monitoring and controlling these complex environments.

Research activity over the past several years has considered the fault diagnosis abilities of humans in a supervisory control context [1], although much of this work has been confined to the process control domain. Of additional importance is the consideration of human performance in monitoring and controlling large scale hierarchical networks, such as communication or command and control systems. These systems typically can be represented as large queueing networks, with the extent of control increasing with successive hierarchic levels. Due to the enormous size of the system, not all relevant information can be displayed to the human operator at one time; thus, multi-page computer generated displays are frequently used. Human limitations in dealing with systems of this type have not been investigated to any great extent [2].

The work reported in this paper is an effort to relate aspects of system design to the complexity of the human operator's monitoring and control task. Emphasis is placed in the following section, therefore, on identifying a variety of perspectives on complexity. A simulated large scale system (an extension of the one reported in Henneman and Rouse [2]) is then described, which is used in an experiment to evaluate two dynamic measures of task complexity that are based on the structure of the system and the strategy of the human operator.

BACKGROUND

The purpose of this section is to review discussions and investigations of complexity that have taken place within a number of disciplines. Computer scientists, for example, are often interested in the computational complexity of a particular algorithm. Computer scientists also often measure the complexity of a piece of software. General systems scientists postulate theories about the inherent complexity of large scale systems, while theoretical biologists discuss the complexity of biological systems. Psychologists relate the complexity of symbolic or spatial patterns to human behavior. Man-machine systems engineers are interested in system complexity as it relates to human problem solving and system control. In this section, the issue of complexity is addressed from these and several other perspectives. For organizational purposes, non-behavioral perspectives are considered first, followed by behavioral complexity perspectives.

Non-behavioral perspectives

Computational complexity. An issue that has interested computer scientists, operations researchers and others is that of the relative computational difficulty of computable functions (i.e., why is one function more difficult to compute than another?). In general, computational, combinatorial, or algorithmic complexity is defined as the length of time or amount of space (memory requirements) required to compute a certain function on a certain type of machine [3,4]. Algorithms are classified in terms of the amount of time (e.g., polynomial or exponential) and/or memory they take to be solved on a computer [5,6,7]. Examples include an analysis of a graph theory algorithm for cluster analysis [8], a consideration of the complexity of mathematical models in manipulator control systems [9], some observations regarding the complexity of matrix factorization [10], and an examination of the time required to solve problems in a system of communicating sequential processes [11].

As Rouse and Rouse [12] have noted, a relatively large amount of work has been done to analyze the complexity of automatic fault detection algorithms. Fujiwara and Kinoshita [13], for example, analyze several problems of instantaneous and sequential fault diagnosis of systems. They show that these algorithms are polynomially complete (i.e., they can be solved in polynomial time if and only if the traveling salesman problem, knapsack problem, etc., can be solved in polynomial time.) Priester and Clary [14], using results from system identification theory, develop measures of failure test complexity. Rouse and Rouse

[12] try to relate human performance to an optimal solution of a fault finding task.

Software complexity. Somewhat related to the measurement of computational complexity is the measurement of software complexity. While computational complexity estimates the time and memory requirements of implementing a particular algorithm on a computer, software complexity estimates such quantities as programming time and program length. By controlling the software complexity, production costs should reduce while overall software quality should increase [15].

Halstead [16] has proposed a theory of software science that is based on a measure which counts the number of operators and operands in a program in order to estimate program length, volume, program level, language level, programming effort, and programming time. Despite a high degree of predictive power, criticism has been leveled at the approach from a theoretical perspective [17,18]. Other approaches include a graph-theory based measure of McCabe [19], an information theory based measure [15], and a control structure/flow measure [15]. Davis [20] notes that none of these approaches are based on a satisfactory model of programmer cognitive processes, and thus, proposes and evaluates measures based on "chunks", or related program concepts that can be understood by programmers as a single cognitive unit. Chaudhary and Sahasrabuddhe [21] conclude on the basis of experimental results that complexity not only involves the control structure of a program but also the executional difficulty of the program.

Complexity of physical systems. Besides the complexity of mathematical algorithms or computer software, complexity has also been discussed in the context of a physical system. Typically these

investigations are of a general, theoretical nature, although some of the discussions are applicable to the consideration of human performance in large scale systems. In the following paragraphs, the general systems approach to understanding complexity is considered.

Weaver [22] has distinguished between problems of simplicity, disorganized complexity, and organized complexity. Problems of simplicity include the largely two variable problems considered by the physical sciences before 1900. Problems of disorganized complexity contain a very large number of variables, each of which may possess an erratic or unknown behavior. By applying techniques of probability theory or statistical mechanics, the behavior of the system as a whole may be analyzed and characterized by its average tendencies. An important range of problems lies between the extremes of simplicity and disorganized complexity. These problems may contain a relatively large number of variables; however, they also exhibit a high degree of organization. Problems of organized complexity are ones in which "a sizeable number of factors ... are interrelated into an organic whole" [22]. In general, these problems are of interest to the system scientist. Systems of all types -- biological, social, economic, ecological, or physical -- can be characterized as highly interrelated subsets of variables.

Redundancy and complexity. A recurrent theme throughout the literature is the identification of system size and degree of interconnectedness as indices or attributes of system complexity. Example domains include general systems [23,24], architectural design [25], and political systems [26].

A highly connected system is complex, however, only in the sense that it is difficult for a person to understand the causal net of relations among system components and variables. Thus, a high level of connectivity (or redundancy) should lead to increased difficulty in solving problems related to system operation (i.e., failure detection, resource management, etc.). Waller [27], for example, proposes that large, highly connected systems are complex and difficult for humans to understand because of inherent human information processing limitations.

With respect to normal system control, however, the concept of redundancy has quite different implications. Mackinnon and Wearing [28] investigated a complex decision making environment in which the number of elements in the system, the degree and pattern of interconnections in the system, and the presence/lack of uncertainties in the system were varied. The results indicated that the complex (or highly interconnected) systems *did not always lead to poorer levels of* performance. In these cases, therefore, a high level of redundancy led to improved system performance. The authors claim that this effect is due to the insensitivity of highly redundant systems to faults and mistakes made by subjects.

Thus, at least two different interpretations of the relationship between redundancy and system complexity exist. The first interpretation, generally espoused by social scientists and general systems theorists, is related to the difficulty of understanding the system. When a failure occurs it may be difficult to locate its cause due to the presence of multiple paths through the system. On the other hand, when a system is highly redundant, its ability to carry on normal operation is greatly increased — the redundancy serves to stabilize the

network. This interpretation is largely held by biologists and engineers. Thus, the level of interconnectedness in a system affects the level of two types of complexity: problem solving complexity and system control complexity.

These two interpretations are consistent with standard results from reliability theory [29,30,31]. As the number of alternate paths (or components) increases in a system, the reliability increases, as expressed by the mean time between failures. However, data has shown that as the redundancy (and hence, the reliability) increases, the maintainability of the system decreases, i.e., the mean time to repair increases [30]. Thus, more complex (or redundant) systems lead to longer repair times. The availability of the system (or the probability that the system is operating satisfactorily at any point in time) is shown by von Alven [31] to be a function of both reliability and maintainability; thus, it too is a function of system redundancy.

Subjective nature of complexity. A final theme within the complexity literature is that of the relative or subjective nature of complexity. Ashby [32] illustrates this concept by considering a sheep's brain. While the internal mechanisms of the brain are very complex to a neurophysiologist, a butcher only has to distinguish a sheep's brain from about 30 other cuts of meat (or about 5 bits). Several other authors also equate complexity with descriptions of objects, rather than with intrinsic properties of objects [33,34,35]. This perspective leads quite naturally to the discussion of behavioral complexity which is pursued below.

Summary. The following conclusions can be made on the basis of the review so far:

1. Complexity is related to the size of the system as well as the level of redundancy (or connectivity) among components.
2. The effects of redundancy on complexity differ depending upon one's perspective. A highly redundant system may lead to better overall performance; however, it may also lead to increased human problem solving difficulty.
3. Complexity can only be measured relative to a person's understanding of the system.

Behavioral complexity

The preceding discussion has made only oblique reference to human abilities in perceiving information about the system or in solving problems within the environment created by the system. From a psychological perspective, the relationship between complexity and human performance is of fundamental importance. This relationship is explored in the following sections. Perceptual complexity is considered first, followed by problem solving complexity.

Perceptual complexity. Rouse and Rouse [12] describe studies of perceptual complexity as dealing with "... the human's ability to recognize, rotate, reverse, etc. displayed patterns as a function of various attributes of the pattern, including number of line segments, symmetry, etc." This form of complexity has typically been investigated via some simple experimental scenarios. Greenberg and Krueger [36], for example, use a letter searching task to examine the relationship between task difficulty (in terms of letter orientation and redundancy) and speed of search. Other studies examine such aspects of complexity as color

[37], stress and its relation to a visual discrimination task [38], and relations between visual complexity and verbal associative value [39].

Hochberg and Brooks [40] derive a complexity measure based on the number of angles, number of lines, and the variety of angles contained within a drawing. Vitz and Todd [41] also propose a complexity metric of non-representational shapes based on a sampling of elements in the drawing. Butler [42] extends this work by using a complexity measure based on information load and the number of lines in the drawing. Attneave [43] develops a complexity measure based on the physical characteristics of shapes. Kimchi and Palmer [44] relate the number of elements in a drawing and its size to subjects' similarity judgements and their verbal descriptions. Finally, Simon [45] reviews several different approaches to relating the perceptual complexity of patterned sequences of symbols to human behavior. Simon concludes that all of the theories share a common central core: subjects perform the tasks by inducing pattern descriptions from the sequences. These descriptions all involve the same rules between symbols, iteration of subpatterns, and a hierarchic phrase structure.

Relative to the role perception plays in the complexity of fault diagnosis tasks, Rouse and Rouse [12], in their study of complexity measures of fault diagnosis tasks, use the number of displayed components as a measure of perceptual complexity. Results indicate that this measure is not a good predictor of fault diagnosis performance. Since the number of components displayed on the screen is a function of the equipment's inherent complexity, not peculiarities of the display, the authors advise that a systematic variation of display characteristics might indicate that fault diagnosis tasks can be perceptually complex.

In light of the success of other predictors which are more related to problem solving complexity that are discussed in the next section, the authors suggest that problem solving measures are more relevant to fault diagnosis tasks.

Brooke and Duncan [46] extend the work of Rouse and Rouse [12] to examine explicitly the effect of display formatting on measures of the fault diagnosis process. Results indicate that changing some of the perceptual characteristics of the display improves the speed and diagnostic efficiency with which faults are located.

Problem solving complexity. A second form of behavioral complexity, which has received less attention than perceptual complexity, is problem solving complexity. This type of complexity measure assesses various problem attributes and attempts to relate them to human reasoning abilities and problem solving skills. Experimental assessments of problem solving complexity typically use syntactic or arithmetic problem solving tasks. Glover et al. [47], using a written learning task, finds that more difficult tasks result in higher levels of recall. McDaniel [48] reports that syntactically complicated sentences result in greater recall of sentence structure than do simple sentences. Ashcraft and Stazyk [49], using mental arithmetic tasks, discover that reaction time increases with increasing problem complexity. Loftus and Suppes [50] find that problem solving difficulty of arithmetic word problems is related to problem attributes like surface structure, number of words, and the number of different operations required to obtain a solution. Morgan and Alluisi [51], using a code transformation task, find that problem complexity has a greater effect on performance after practice than the early trials.

Kieras and Polson [52] discuss "user complexity," which is the complexity of a device or system from the point of view of the user. The authors propose that user complexity depends on the "amount, content, and structure of knowledge required to operate a device." In addition, the complexity for a novice increases as a function of the difficulty of acquiring that knowledge. Knowledge is composed of two components, task knowledge and device knowledge. Complexity, therefore, is dependent not only on device or task characteristics, but also on the knowledge of the user. In order to measure complexity, the authors suggest the following indices: number of productions (rules) to be learned, number of productions fired, number of keystrokes, number of items in working memory, etc. The authors propose that these measures of user complexity can be determined by using a computer simulation to implement a user model.

With respect to measures of problem solving complexity in man-machine systems, the most pertinent work is that of Rouse and Rouse [12]. Besides their measures of number of components and optimal solution which have already been discussed, Rouse and Rouse also propose two measures of problem solving complexity: the number of relevant relationships (i.e., number of possible causes of a set of symptoms) and an information theoretic approach. These two measures are highly correlated with human performance in the fault diagnosis tasks (as measured by time to solution). The authors suggest that the success of these measures can be largely explained by the fact that they reflect the human's understanding of the problem and his resulting solution strategy.

Wohl [53,54,55] examines the relation between the structure of electronic equipment and human fault diagnosis performance. He derives a

measure of complexity based on system connectivity which is shown to predict repair times very well. Wohl relates this measure to human cognitive limitations. He suggests that if some upper bound of complexity is reached (namely, human short term memory limits), some fraction of equipment failures will be non-diagnosable. Existing equipment does not exceed these human cognitive limits since designers as well as diagnosticians possess the same limits. However, these results have rather important implications for computer-aided design, which could allow the creation of overly connected parts. It should be noted that although this measure is related to the Rouse and Rouse measures of complexity, it differs because it reflects mostly characteristics of the system rather than characteristics of the human.

Summary. The following conclusions can be made on the basis of the review of the behavioral complexity literature.

1. Measures of problem solving complexity appear to be most relevant to the task of failure diagnosis, although perceptual complexity may play some part in affecting task difficulty.
2. Complexity is caused not only by the attributes of the problem solving environment, but also by the human's understanding or perception of those attributes.
3. Little work has assessed the complexity of problem solving in large-scale man-machine systems.

Implications of complexity

It is reasonable to assume that complexity should manifest itself in some measurable way; i.e., a complex system should result in longer times to failure diagnosis, longer reaction times, etc. In order to validate a

complexity measure, it is important to identify correctly and to justify an appropriate dependent measure.

A survey was made of 19 behaviorally oriented studies of complexity reviewed in this section. The most popular dependent measure (eight) was reaction time or time to problem solution. Other dependent measures were solution success, recall of sentence structure, memory of forms, and dimensionality judgements of figures. Three studies used number of errors as the dependent measure. Few of the studies, however, (other than Rouse and Rouse [12]) offer any rationale for their choice of a dependent measure.

Conclusions

The preceding sections have considered definitions, measures, and implications of complexity within a variety of domains. On the basis of this review, it is instructive to make some generalizations.

Most studies of complexity performed by systems scientists are on a general level. Although much work has gone into defining and measuring system complexity, little has been done to assess the implications of complexity. Furthermore, assuming that humans must play an important role in many large scale systems (e.g., failure diagnosis and network management), little research has investigated the relationship between large scale system complexity and human performance. Due to the strong theoretical flavor of this approach, it is often difficult to see its application to real world systems.

On the other hand, studies of complexity performed by behavioral scientists are on a very applied level. Although the approach often lacks the theoretical rigor of the systems approach, complexity is always

related to some aspect of human performance. Unfortunately, differences between tasks and complexity measures make it difficult to generalize results across contexts. Moreover, the small, well-defined nature of the tasks seems to have little relation to human performance in large scale systems.

The remainder of this paper is devoted to consideration of human performance in monitoring and controlling large scale systems. Thus, the research attempts to integrate a number of the issues raised in this section concerning the nature of complexity. Complexity is viewed as being a result of both the structure of the system and the human operator's understanding of the system. Complexity is also considered in terms of its relation to both system performance and human performance. In particular, the relationship between such structural variables as redundancy and number of levels and performance is investigated. In summary, the goal of this work is to "bridge the gap" between systems science and behavioral science and, in the process, gain practical insights into appropriate roles for humans in the increasingly complex systems that technology is producing.

TASK DESCRIPTION

A previous experiment [2,56] considered human performance in the monitoring and control of an essentially context-free representation of a large scale system. Subjects monitored and controlled a computer simulated large scale system called MABEL (Monitoring, Accessing, Browsing, and Evaluating Limits), trying to optimize such system parameters as number of customers served and customer processing time while trying to diagnose system failures. As noted in the Background

section, of interest is the assessment of measures of task complexity; i.e., what features of the physical system, the human-system interface, or the human's understanding of the system make the monitoring and control task difficult? A major goal of this paper is to consider the nature of complexity in a large scale system.

The remainder of this section describes a contextually augmented version of MABEL that contains substantially higher fidelity than the earlier simulation. An experiment is then described, from which data are analyzed using the same set of performance measures as were applied to the experiment reported in Henneman and Rouse [2]. Data are then analyzed from the standpoint of assessing task complexity.

Overview of CAIN

Certain features of MABEL were substantially changed to develop CAIN (Contextually Augmented Integrated Network); however, the underlying structure of CAIN is identical to that of MABEL. This section summarizes the similarities between the context-free MABEL and the contextually-augmented CAIN. The summary is only a very broad overview; the reader is referred to Henneman and Rouse [2] or Henneman [56] for much more detail concerning the underlying structure of the two simulations.

CAIN is programmed in Pascal on a VAX 11/780 computer and operates in real time. It is structured as a large hierarchical network that can range in size from hundreds to thousands of nodes. Customers travel through the system from a randomly selected source node to a random destination. Subjects monitor this system activity via a CRT display. When they detect a problem in the system (possibly due to a failure), subjects issue an appropriate command through a keyboard to correct and

compensate for the abnormal situation. The overall objectives of the operator are:

1) to maximize the number of customers served, and
2) to minimize the time it takes for customers to travel between source and destination nodes.

Because there are so many nodes in the network, it is not possible to display information about all nodes at one time. Thus, nodes are grouped into relatively small networks called clusters. Human operators are restricted to viewing only one cluster at a time on the CAIN display. Clusters are grouped into hierarchic levels.

Effects of Node Failures

Under normal circumstances, CAIN operates automatically without interference from the human operator. Since the system cannot automatically diagnose and repair failures, the human must monitor the system looking for evidence of failed components. Node failures can occur in two ways. The first is a randomly occuring failure mode caused by malfunctioning equipment. The second type, capacity failure, can be caused by the randomly occuring failures. Each node has a maximum number of customers that it can store at one time. If this limit is exceeded, the node fails. Thus, if a node fails randomly and a customer needs to visit that node, it will be retained at its previous node. This retention will cause the previous node to stop processing customers, which can lead to a capacity failure. In this way, if the operator does not quickly locate failures, the problems will propagate through the system.

Addition of Context

Although the physical hierarchical structure of MABEL was preserved, the addition of contextual information to CAIN required changing some interface characteristics. In the MABEL scenario, for example, all nodes on a display page are identified by a number on the CRT display. Each displayed node in a cluster, therefore, is physically identical to nodes in other clusters. The MABEL interface has a generic quality in that all subsystems are visually similar; no contextual cues exist. On the other hand, nodes in CAIN are identified via specific geographic locations. Thus, a node in MABEL with the label "9" might be labelled "Chicago" in CAIN. A typical CAIN display is shown in Figure 1.

Simply introducing geographic names as node labels is not enough, however, to alter subject task performance. A small experiment (n=3) replicated the first MABEL experiment [2,56], with the exception that nodes were given geographic names. Subjects still referred to nodes by number only; contextual labels were present but not needed to perform the task. No significant difference was found in terms of performance between subjects using the two task scenarios. This result suggests that the addition of context must be such that it provides associative links (i.e., memory aids) or cues (i.e., clues to the location of problems within the system) through which subject performance is enhanced or task difficulty is decreased.

Associative Links. The formation of associative links in CAIN is facilitated by the way in which a subject identifies a node. In CAIN, nodes are referred to by geographic labels only, never by number. Subjects may input the shortest string of characters that uniquely

identifies the node from all other nodes in the system. Thus, "Denver" may be abbreviated "den". Most nodes can be identified with a three or four character substring of the complete name. In addition, the number of elements on a display page is kept constant at 16 so that the contextual information is invariant.

To illustrate the effect this change has on the subject's task, consider the command that displays a lower level cluster. In MABEL, the subject inputs the command "d2", which displays the cluster beneath Node 2. In CAIN, on the other hand, the subject types "dSanf", which displays the cities beneath San Francisco (e.g., Berkeley, San Jose). Thus, subjects can form associations or links between system parts due to the existence of contextual information.

Subjects can use these learned associative links to maneuver through the CAIN display hierarchy. In MABEL, movement between display pages is constrained to the cluster of nodes immediately above or below the current display. Thus, it is not possible to jump laterally across the network. In CAIN, however, it is possible to move from one part of the system to any other part. For example, if a subject recalls that the cluster associated with Bangor, ME was previously experiencing problems, it is relatively easy to call up that cluster display. This is done by using a "find" command ("f"). In addition, subjects can return immediately to the highest level in the system by inputting the "a" command. (A complete list of commands available for use in CAIN may be found in Table 1. This command list is categorized by function: access, monitor, diagnose, or control.)

Cues. The formation of cues in CAIN is provided by the introduction of context-dependent events. These events are of one of two types:

recurring failures and non-uniform loading. Although equipment in nodes fails randomly, some equipment experiences a higher probability of failure. For example, a thunderstorm in Little Rock, AR may make equipment in that city susceptible to lightning damage. Similarly, given that incidents of vandalism are more likely to occur in Newark, NJ than in Council Bluffs, IA, there is a greater chance of equipment damage in Newark. Therefore, equipment in certain cities exhibits a greater tendency to fail than in other cities. Subjects are informed of these locations via warning alarms that appear on the bottom of the display. Subjects can directly monitor activities within these trouble spots via a special "watch" ("w") command. Subjects acknowledge the alarms by inputting an "erase" command ("e"). Subjects add and delete trouble areas from the watch list by using "+" and "-" commands.

Besides recurring failures, another type of context-dependent event present in CAIN is non-uniform loading. At different times, certain sections of the system may be prone to experience heavy loading. For example, certain times of day are busier in one part of the country than in others. Similarly, a major political or sports event in one section of the country may increase the number of messages sent. As with the recurring failures, subjects are told the location of these increased loads via a message at the bottom of the screen. Subjects can reduce the number of customers admitted to the overloaded subsystem by means of the "load" ("l") command.

In summary, despite the structural isomorphism of the two simulations, CAIN represents a significant departure from the context-free scenario of MABEL. Through the addition of contextual detail and the addition of events that are dependent upon this contextual

information, the simulation fidelity has been increased significantly.

MEASURES OF COMPLEXITY

The Background section considered complexity from non-behavioral and behavioral perspectives. When assessing the complexity of an operator's task in monitoring and controlling a large scale system, both approaches should be taken into account. In this paper, therefore, the complexity of a large scale system is described in terms of: 1) the physical structure of the system and, 2) the operator's understanding of the system as reflected by his strategy. From this perspective, a system that is complex or difficult to control for one operator may be relatively easy to control for another operator. Similarly, the complexity of a system may vary with time for any particular operator. Some systems, however, may be complex regardless of any particular control strategy due to their inherent structural complexity. The following paragraphs propose two measures of complexity that incorporate these ideas. Structural complexity is considered first, followed by strategic complexity.

Structural Complexity

A one-to-one relationship exists between the hypothetical physical structure of CAIN and the actual structure of the display page hierarchy. Since the main control task in CAIN is to locate failures, a measure of structural complexity should assess the difficulty of finding failures given the physical arrangement of the system. A major constraint placed on an operator's ability to locate failures is the hierarchical display structure; thus, it seems reasonable to assert that structural complexity

can be estimated by calculating the total number of display pages the operator must view in order to repair all system failures. Assuming that the operator knows the location of all failures, this measure represents the minimum number of pages necessary to locate all system failures. Thus, the structural complexity measure represents optimal performance given the constraints of the structure or arrangement of the system components. Operator performance affects this measure only in that any particular operator may have more or fewer failures depending upon his fault finding ability.

To illustrate how this measure is calculated, consider the system in Figure 2. This hypothetical system contains four nodes per display page and has three levels. Each group of four rectangles represents a cluster of nodes (i.e., one display page). For clarity, only those clusters of nodes that enter into the complexity calculation are shown. The darkened rectangles represent nodes that have failed. In this example, three failures exist within the system: two on the second level and one on the third level.

The structural complexity measure is determined by counting the number of display pages that must be viewed in order to find all failures. The counting method assumes a strategy based on tracing higher level symptoms to their causes in the lower levels. (Context-specific cues might, of course, allow operators to locate failures in fewer pages.) Thus, the counting method assumes that after locating all failures along one subsystem branch, the subject returns to the highest system level to search the next branch (a depth-first strategy). Figure 2 is self-explanatory; to repair all three failures in the system, an operator must view at least six display pages. The final return to the

top system level is not counted into the measure because it would simply add 1 to all estimates.

Strategic Complexity

The strategic complexity measure explicitly considers operator performance. When an operator is deciding which path through the system is most likely to lead to finding a failure, he makes a tradeoff between his uncertainty concerning the state (i.e., queue lengths) of a subsystem display page and his expectations of finding a failure in that subsystem. High uncertainty about a subsystem may be acceptable, for example, if a relatively low probability exists of finding a failure on that display page. On the other hand, high subsystem uncertainty may be unacceptable if a very high probability exists of finding a failure. These observations suggest that an appropriate measure of strategic complexity that reflects the trade-off between state uncertainty and probability of failure is the multiplication of these two metrics.

State uncertainty (U) is defined as the real time elapsed since a particular display page was last tested for failures. Probability of failure is defined as the probability that a failure exists within a cluster given the state of the display ($p[F|X]$). For example, when a subject views a particular display page, features of that display provide information about the existence of failures in other subsystems (e.g., a large queue size suggests a lower level failure.) Experimental data files were replayed in order to estimate these probabilities empirically. These probabilities were determined by dividing the frequency with which a display state reflected a failure by the frequency with which a particular display state was viewed by an operator. Sets of

probabilities were calculated for different system configurations (2 vs. 3 levels and high vs. low redundancy), and different loading rates (e.g., a system with a low loading rate has fewer customers in service, and hence, lower queue sizes will reflect failures).

The measure of strategic complexity multiplies these two measures (state uncertainty and probability of failure given the system state) and sums the product across all clusters in the system:

$$\text{Strategic Complexity} = \sum_{i} U(i) \times p[F|X(i)]$$

where $U(i)$ = time since last accessing display page i

$X(i)$ = State of page i reflected by display one level higher

$p[F|X(i)]$ = probability of failure given state i

and F denotes "failure"

When a subject descended to a lower level, the $p[F|X(i)]$ remained fixed for the previous level. When a subject returned to the higher level, the $p[F(X(i)]$ values associated with the just-visited lower level cluster were set to zero. Thus, when an operator descended to a lower level subsystem and tested for failures, the strategic complexity measure was simultaneously increased by the "new" uncertainty present in the other lower-level subsystems and decreased by the certainty now associated with the current level.

To illustrate how the strategic complexity measure is determined, consider the display in Figure 3. This system contains four nodes per display page and has two levels. The operator is viewing the highest

level page in the display hierarchy and is monitoring activity in the next level of the system. The operator can gather information about activity in the second level of the system from two sources in this example: the cluster display and the data displayed via the monitor command. The monitor command lists the number of customers in the clusters one level below; the cluster display shows the number of customers waiting at all nodes in the current cluster.

Each of these pieces of information reflects the probability that a failure has occurred in a lower level cluster. These probabilities (which are plausible, but hypothetical) are listed in Table 2. For example, the queue size of 15 in Denver reflects a relatively high probability (0.75) that a failure exists in Level Two. Similarly, the monitor command reports that eight customers are currently in the cluster beneath Denver; these eight customers reflect a 0.60 probability that a failure exists. The operator has not tested the cluster beneath Denver for failures for U(Denver) = 20.12 seconds. Using the information that reflects the highest probability of failure (i.e., from the cluster display) results in the following measure of strategic complexity for the Denver region:

$$U(\text{Denver} \times p[F|x(\text{Denver})] = 20.12 \times 0.75$$
$$= 15.09\text{s}$$

This procedure is then repeated for the other clusters in the network and the measures are added together. In this way, the total strategic complexity is determined to be 15.61.

In this example, it should be noted that Denver makes a very large contribution to the strategic complexity measure as a result of two factors: first, the operator has a high degree of uncertainty concerning the Denver subsystem in that he has not tested that cluster for failures in 20.12s. Second, the display reflects a very high probability (0.75) that a failure exists in the Denver subsystem. The combination of these two factors leads to a very high measure of strategic complexity for the Denver subsystem. On the other hand, the other subsystems have either a low uncertainty measure or a low probability of failure. Thus, their contribution to strategic complexity is small.

Finally, it is instructive to consider the extent to which an operator may "optimally" reduce strategic complexity. Since the measure is based on time, it will continually increase unless either the operator performs some action or the state of the system shifts. At any instant in time, therefore, it is possible for an operator to reduce strategic complexity optimally by viewing the display page that reduces the measure by the largest amount (i.e., the cluster with the largest $U \times P[F|X]$ value). In the long run, however, the measure may only be optimally reduced given the operator's performance constraints (i.e., psychomotor reaction and movement times). In other words, since the measure will in general keep increasing with time, optimal performance will always be limited by how long it takes the operator to physically select the next display page.

Dependent Measure of Complexity

The literature review also suggested that an appropriate dependent measure of complexity is the time until failure diagnosis. In the

context of CAIN, this measure is the average time until the subject issues a repair command for a failed node. Since the two independent complexity measures vary with time, it was necessary to use a dependent measure that also changes with time. Average time, therefore, includes the diagnosis time for the current repair plus diagnosis times for the four previous repairs.

Summary of Complexity Measures

To summarize, the structural measure reflects an inherent characteristic of the network, namely the number of display pages necessary to find all of the failures in the system. The strategic measure, on the other hand, reflects temporal aspects of subjects' strategies, i.e., subjects' paths through the network. From this perspective, the strategic measure reflects the complexity resulting from a particular strategy.

Although the two complexity measures proposed here may have some general applicability (in particular, the measure of strategic complexity is appealing due to its temporal nature), it is not the intent of this paper to suggest or prove that these measures are true indices of task complexity. The goal instead is to show in a pragmatic sense that these two dimensions represent a useful distinction relative to task complexity. These measures represent a convenient means to demonstrate this distinction.

METHOD

Motivation

The main goal of this experiment was to investigate the nature of complexity in a large scale human-machine system. As emphasized in the preceding section, the general assumption is made that task complexity can only be measured relative to an individual's understanding of the system and his expertise in dealing with problems in that system. Thus, complexity is considered to be dynamic, varying across time and among subjects. Accordingly, as discussed below, subjects were required to perform the task (CAIN) over a relatively long period of time.

Subjects

Eight junior and senior engineering majors at Georgia Tech served as subjects in this experiment. Due to the nature of the task, potential subjects were screened via a typing test (minimum ability level was 25 words/minute). Subjects were paid a total of $65: $5.00 for each training session (3) and each experimental session (10).

Training

Subjects were trained via a combination of written instructions and hands-on experience with CAIN. Subjects initially were given two sets of written instruction on consecutive days explaining the system, the goals of their task, and methods for achieving these goals. Self-test questions were contained within the text to insure mastery of the material. The experimenter reviewed this material with subjects at the beginning of each training session. In addition, subjects were given one-page summaries detailing the structure of the system, available commands (Table 1), and operation of the system.

Subjects completed the first two training sessions by controlling a two-level CAIN system. The third training session was spent controlling a three-level CAIN system. These sessions were performed using a version of CAIN that allowed subjects to start and stop the program execution. Thus, subjects could investigate normal and abnormal system functions without being overwhelmed by the progressive effects of failures. The experimenter was present during all training sessions to answer questions.

Experimental Design

Henneman and Rouse [2] reported that cluster size (number of nodes per display page) in MABEL had a particularly strong effect on task performance. Results suggested that small clusters degraded performance because fewer connections existed between nodes; less redundancy caused failures to propagate more quickly. Another result from Henneman and Rouse [2] showed the very strong effect of number of hierarchical system levels on human performance. Increasing the number of levels from two to three degraded performance. Thus, two independent variables selected for further analysis were the degree of redundancy (or connectivity) and the number of levels in the system. (Cluster size was kept constant at 16 as mentioned previously in order to emphasize the non-varying features of the contextual display.) Redundancy or connectivity was defined as the number of connections emanating from each node. Redundancy varied between low (6 connections/node) and high (13 connections/node) and number of levels varied between two and three.

Of interest in this experiment was the way in which complexity changes as subjects gain expertise. Thus, the order of presentation of experimental conditions was not randomized. All subjects saw the same

experimental conditions in the same order. A final independent variable, therefore, was the order of presentation of experimental conditions.

In summary, the ten experimental sessions (S1 - S10) were performed in the following order (with the intent of increasing experimental difficulty): S1,S2: 2 levels, high redundancy; S3,S4,S5: 3 levels, high redundancy; S6,S7: 2 levels, low redundancy; S8,S9,S10: 3 levels, low redundancy. Each experimental session was performed on consecutive days and lasted about 45 minutes.

RESULTS

Summary of Approach

Data from this experiment were first analyzed using the same performance measures as the experiment reported in Henneman and Rouse [2]. Overall results from the analysis of variance supported those of the earlier experiment. In light of this similarity, these general results are only briefly summarized below. Considerably more detail may be found in Henneman [56].

Measures of fault diagnosis performance were affected as expected by the independent variables. Increasing the number of system levels from two to three corresponded to a higher average time to failure diagnosis. This result was largely because failures take longer to propagate upwards in the 3-level systems. In addition, failure-related symptoms take longer to emerge in highly interconnected networks; thus, the high redundancy systems resulted in longer average times to diagnosis.

The fraction of failures repaired by subjects was also significantly affected by increasing the number of levels: as the number of levels increased from two to three, the fraction of failures found decreased

from 0.95 to 0.69. As in Henneman and Rouse [2], subjects could not cope with the very large search space in the three level systems.

Data were also analyzed with the purpose of investigating relationships between the complexity measures, the CAIN environment and operator performance. This investigation was accomplished in two ways. First, an analysis was undertaken of average or global measures of complexity (i.e., the complexity time series averaged over each experimental run). The effect of the experimental independent variables (number of levels and degree of interconnectivity between nodes) on the average complexity measures was determined by using analysis of variance. The relationship between the average complexity measures and measures of subject fault diagnosis performance was then assessed by using correlation analysis. As is discussed below, this analysis of average complexity values provided explanations for differences that exist between different system configurations.

The second way in which complexity was investigated involved using a fine-grained approach, namely, time series analysis. Time series analysis was selected due to the intrinsic time-varying nature of the independent and dependent complexity measures. As will be seen, this analysis provided insight into the way in which complexity evolves and affects different phases of the failure diagnosis process.

Due to the amount of time necessary to perform these analyses, the results are limited to Sessions 2,5,7, and 10. Data for the analyses were generated by replaying subject data files. Following every three seconds (corresponding to the rate of display update), both complexity measures and the average time until failure diagnosis were calculated. Average values for all measures were calculated from these time series.

Analysis of Global Complexity Measures

Analysis of Variance. The results of two ANOVAs (using average structural and strategic complexity measures as dependent measures and number of levels and degree of redundancy as independent measures) are qualitatively summarized in Figure 4. (Henneman [56] reports the results more fully.) Structural complexity, as measured here, may be decreased in two ways: 1) decreasing the number of system levels and 2) decreasing the number of system failures.

The first way (decreasing number of system levels), enables subjects to access fewer display pages in order to diagnose failures in the lowest system level. The second way (decreasing number of system failures) is facilitated by increasing the network redundancy (i.e., increasing the number of connections between nodes). As network redundancy increases, the average number of node capacity failures decreases, which has the effect of decreasing the structural complexity measure.

Strategic complexity, as measured here, may be decreased in three ways: 1) utilizing an effective strategy in terms of responding to symptoms, 2) decreasing redundancy, and 3) decreasing number of levels (which causes symptoms to emerge more rapidly). Subjects tended to trace failures to the lowest system level only when a symptom (i.e., visual cue) appeared on the display, even if they had not viewed a particular region in a large period of time. Consequently, when symptoms emerged slowly (as in the high redundancy/three level conditions), high uncertainty resulted. This uncertainty helped to create moderate to high strategic complexity. On the other hand, symptoms emerged more rapidly in the low redundancy/two level conditions. Since operators tended to

respond primarily to visual symptoms, low redundancy led to low values of strategic complexity.

This dependence on visual cues has implications for the design of task performance aids. Aids should help people to overcome their inability or reluctance to reduce system uncertainty despite the absence of failure symptoms. Alternatively, cues or symptoms could be enhanced so that operators naturally pursue leads sooner.

In summary, increasing redundancy (or number of connections between nodes) led to less structural complexity but more strategic complexity. This result reflects findings from the literature: more redundant systems (corresponding to less structural complexity) enhance the proper operation of the system by reducing the impact of failed components. On the other hand, more redundancy leads to increased strategic complexity (the complexity of failure diagnosis) due to the slower emergence of failure symptoms.

In addition, increasing the number of system levels increased both types of complexity. Again, although multiple system levels might be desirable in that they allow supervision of larger networks and protect upper levels from the effects of failures, they have the undesirable side effect of masking symptoms from operators, thereby increasing the complexity of failure diagnosis. Multiple displays could possibly be used to reduce this complexity.

Correlation Analysis. Pearson product-moment correlation coefficients were calculated between the two average complexity measures and the two dependent measures (fraction failures diagnosed and average time to failure diagnosis). Results are qualitatively summarized in this section; again, Henneman [56] contains more detail. Since significant

interaction effects due to the experimental conditions were found, the analysis was limited to comparisons among correlation coefficients within each experimental condition. Major differences among coefficients were only noted when comparing across the number of levels variable. These results are qualitatively tabulated in Figure 5.

Considering structural complexity first, the measures for both two and three level systems correlate negatively with the fraction of failures found (correlations range between -.31 and -.88). Thus, when many system failures are present on the average (as suggested by a high structural measure), a smaller fraction of failures are found. With respect to the structural measure and average time to failure diagnosis, no significant correlation exists for the two level systems, while high negative correlations (-.63 and -.70) exist for the three-level systems. In other words, high structural complexity in the three-level systems led to shorter failure diagnosis times. This result, being somewhat counter-intuitive, is caused by the following chain of events. High structural complexity is caused by a large number of failures, which are caused, in turn, by a high number of capacity failures. Most capacity failures are located in the upper system levels where failure diagnosis times are relatively short.

The correlations associated with the strategic complexity measure tend to be smaller. Correlations between average strategic complexity and percent failures diagnosed are negative for two level systems (-.34 and -.49) and positive for three level systems (.25 and .49). In the two-level conditions, therefore, high strategic complexity led to fewer diagnosed failures, although in the three level systems, high strategic complexity led to more diagnosed failures. Results for the two level

systems are as expected. High values of strategic complexity resulted from high uncertainty and high conditional failure probabilities. Apparently subjects who used strategies that tolerated these high values were not looking at or using the display cues to find failures; thus, they found few system failures.

Results for the three level systems are less intuitive. Subjects who found many failures in the three level systems had to spend time accessing third level subsystems. Because they spent more time in the third level, these better subjects had to tolerate greater uncertainty about the rest of the system. This increased uncertainty had the effect of increasing the strategic complexity measure.

Summary. In summary, the results presented in this section provide insight to the overall characteristics of the two complexity measures and their relationship to subject fault diagnosis performance. The measures are sensitive to variations among the system characteristics of number of levels and degree of redundancy. In general, the more complex systems have three rather than two levels. The effect of redundancy on complexity depends on the type of complexity: low redundancy networks result in more structural complexity; high redundancy networks result in more strategic complexity.

An important conceptual and methodological issue raised by these results concerns the multidimensional nature of complexity. In particular, the relationship between the independent and dependent measures of complexity is of interest. When many failures exist in a system, the general tendency is for the complexity measures to increase. At the same time, however, the average time to failure diagnosis

decreases. Thus, even though complexity may be large, failure diagnosis time may be small.

This observation emphasizes the distinction mentioned previously between proper system functioning and the complexity of failure diagnosis. In a localized sense, control in a complex system is simple: no matter what the operator does, he will find a problem. This is reflected by short diagnosis times. In a global sense, however, control in a complex system is complex: so many problems exist in the system that proper operation is endangered. This is reflected by a low fraction of failures found. The operator, dealing with only a small part of the system at one time, may be oblivious to the scope of problems in the network. Another important issue is, therefore, the impact of a richly interconnected, multiple-level system (that supports proper system functioning) on the complexity of human monitoring and control (that will degrade failure diagnosis performance).

Analysis of Fine-Grained Complexity Measures

Time Series Analysis. Time series analysis was used to identify, estimate, and diagnostically check transfer functions that relate the two input complexity measures to the average time to failure diagnosis. The general approach is discussed by Box and Jenkins [57]. Each transfer function model predicts the current average time to failure diagnosis through a linear combination of the complexity measures at various time lags. The essence of the modeling process is to determine the time lags to include in the model and the weight or relative contribution of each time lagged variable to the predicted value. Montgomery and Weatherby [58] provide a good tutorial on multiple input transfer function models.

Transfer functions for each subject were developed for Sessions 2,5,7, and 10. (Due to space considerations, these functions are not shown here; the reader is referred to Henneman [56] for more detail.) Overall, the approach was successful. The equations remove all structure from the autocorrelation function of the model residuals. Furthermore, a comparison of the sum of squares of the original dependent time series (i.e., average time to failure diagnosis) to the sum of squares of the residuals shows that the transfer functions explain 82% to 97% of the variance within the original data. Nevertheless, wide differences in the lag and coefficient values in the models exist among both subjects and systems.

The remainder of this section is devoted to the development of a consistent explanation for these differences. The goal is not to account fully for each parameter, lag value, and coefficient in the transfer functions. Instead, the goal is to suggest a plausible explanation for the transfer function characteristics and to suggest reasons for deviations from this explanation.

Explanation for Transfer Functions. The initial step was to identify characteristics of the task, the system, or the human that could explain differences among transfer functions (e.g., long lags and inconsistency of numerical signs). For example, several different events are associated with the life cycle of each system failure: failure occurrence, symptom emergence, and failure diagnosis. Failure occurrence is defined as the time when a part of the system fails; symptom emergence is defined as the time a failure first affects any node that appears on the subject's video display; failure diagnosis is defined as the time a subject issues a repair command for a failed component. The timing of

these events undoubtedly has some effect on the length of time needed to find the failure. Moreover, the system complexity at these event times might also affect failure diagnosis time.

Besides the possibility that different events associated with the failure life-cycle impact diagnosis time, it is also reasonable that different types of diagnosis might affect failure diagnosis time. The diagnosis of any particular failure may be classified into one of three types: topographic, symptomatic, or serendipitous. Subjects identifying failures using a topographic strategy trace failure symptoms from higher system levels to their causes in lower levels. Subjects identifying failures using a symptomatic strategy make a direct mapping from their knowledge of the system structure to the failed component. A symptomatic diagnosis relies, therefore, on the subject's contextual knowledge of the system. For example, when subjects make a jump from one cluster to another cluster in the same level to repair a failure, their action suggests that their context-specific knowledge of the system is providing guidance to system trouble areas. Finally, subjects may also identify failures accidentally or serendipitously. In this diagnosis mode, subjects locate failures while browsing through the system or while tracing the cause of a different failure.

In summary, it is possible that several different types of failure-related event (e.g., failure occurrence and symptom emergence) and several different modes of failure diagnosis (e.g., symptomatic, topographic, and serendipitous) can affect the time to failure diagnosis within a system. In addition, due to the aforementioned aggregation of five failure diagnosis times for the dependent complexity measure, it is possible for many lags (possibly quite long) to enter into the transfer

functions. From the perspective offered in the preceding paragraphs, therefore, the transfer functions relating the two complexity measures to failure diagnosis time are affected not only by system characteristics and individual differences; rather, the equations are also affected by types of failure-related event, modes of failure diagnosis, and the way in which diagnosis times were aggregated. In the next section, these factors are considered analytically and compared with the transfer functions.

Empirical Analysis. Given the preceding discussion, subject data files were replayed[1] in order to gather failure-related event information. When a subject repaired a failure, it was classified as being topographic, symptomatic, or serendipitous by using the following heuristics. A failure diagnosis was classified as topographic if the failure was affecting the last higher display page viewed by the subject; the assumption was made that the subject was tracing the cause of symptoms via the physical structure of the system. A diagnosis was classified as symptomatic if the subject jumped more than one level in the display hierarchy, if the subject jumped laterally on the same level, or if the subject diagnosed the failure on the basis of contextual messages. All of these instances suggested that the subject was using contextual knowledge of the system to recall the likely location of failures. Finally, a diagnosis was classified as serendipitous if no

---

[1]The comparisons made in this section are limited to data from Session 2. Since the major goal is to show how the results that did arise are explainable, the explanation can be accomplished by examining only a subset of all the data.

symptoms existed on the previous level and for the second, third, etc. failures diagnosed on a single display page.

Failure occurrence and symptom emergence times were also determined. This information was collected for each diagnosis mode (i.e., topographic, symptomatic, and serendipitous) and also aggregated across all diagnosis modes. Using these data, the average time from each event type to the time of diagnosis was calculated. A comparison of these average times to the transfer functions lag values for Session 2 may be found in Table 3. Table 3 may be interpreted as follows. For each subject (1-8) the total number of failures repaired for each diagnosis mode are listed along with the fraction of the total for each mode. The top row in each pair of boxed numbers corresponds to average event times that are approximately equal to lag values from the transfer functions. The lower row in each box contains information about the corresponding transfer function variable. The + or - represents the numerical sign of the transfer function coefficient, "struct" or "strat" refers to the type of complexity, and the final number is the time lag value.

Table 4 presents some of the information in Table 3 in a slightly different form, listing only the empirical average time values paired with transfer function lag values. As Table 4 clearly shows, a very high degree of correlation exists between the time values and the lag values ($r = 0.92$, $p < 0.01$).

Several patterns are evident in Table 3. First, of the eight subjects, seven have transfer function lags that are approximately less than or equal to the overall average time to failure diagnosis (all except Subject 8). Four of these lags involve a strategic complexity component (Subjects 2, 4, 5, and 7), and four involve a structural

component (Subjects 1, 3, 6, and 7 — Subject 7 has both). Furthermore, the numerical sign of the transfer function coefficient in each case is positive. In each case, therefore, the strategic complexity measure is related positively to the predicted failure diagnosis time.

This finding is intuitively plausible. The measure of strategic complexity reflects the trade-off between the subject's system state uncertainty and the probability of failures existing within the system. If this measure is at a relatively high level when a failure occurs, the time needed to find that failure will be increased due to the number or severity of potential problem areas within the system. A high measure of strategic complexity suggests that many subsystems (clusters) have potential problems and thus, require the attention of the operator. The time necessary to observe these clusters has the cumulative effect of increasing time to failure diagnosis.

Similarly, the measure of structural complexity estimates the minimum number of pages that the subject would have to view in order to find all system failures. On the average, the time needed to locate any one failure in the system will increase as this measure increases.

A second pattern that exists within these results is the similarity between the average time from symptom emergence and the lag values associated with a negative structural complexity component (Subjects 2, 3, 4, 6, and 8). After a symptom emerges, therefore, the structural complexity measure decreases the predicted time to failure diagnosis: the greater the structural complexity of the system, the less time it takes to locate failures. This counter-intuitive result may be explained as follows: as system structural complexity increases, more failures exist within the system. As the number of failures in the system

increases, it is likely that a subject will locate some failures rather quickly.

This observation reflects the relation between fault diagnosis time and number of failures in the system. As the number of failures in the system increases, one might expect the time to diagnosis for some failures also to increase. On the other hand, as the number of failures in the system increases, the chances of finding some failures fairly soon is relatively high. Thus, once a symptom emerges, the average time to failure diagnosis will decrease simply because there are more possible failures in the system to find.

This conclusion is consistent with the relation between lag values and the calculated average time between symptom emergence and failure diagnosis for serendipitous diagnoses. Four of the eight subjects (Subjects 2, 3, 4, and 8) have a negative structural complexity component that is related to the symptom emergence for this mode of failure diagnosis. This increase/decrease effect of complexity, therefore, appears to be dependent upon both the type of complexity (structural or strategic) and the type of failure-related event (e.g., failure occurence or symptom emergence).

In one situation, the measure of structural complexity at the time of symptom emergence appears to increase failure diagnosis time (Subject 7). It is worthwhile noting that this subject located substantially more failures than any other subject (97 vs. 75 -- the next highest total). This high number was not due to an effective strategy; rather, the subject used a poor strategy that resulted in a very high number of capacity failures -- note the high percent of serendipitous failure locations relative to the other subjects. Indeed, all of the

coefficients in Subject 7's transfer function (Table 4) have positive coefficients. In short, the subject was unable to overcome the number of failures in the system, thereby resulting in an increasing time to failure diagnosis.

So far the discussion has centered on the aggregated mode of failure diagnosis. Examining the individual modes of failure diagnosis, similar trends are apparent except in one case: the transfer function associated with topographic diagnoses have no lag values that correspond to any of the inter-event times. What characteristic of topographic diagnoses could cause this lack of association? One possible reason is simply that there are fewer topographic diagnoses made by subjects; thus, it is less likely to obtain an accurate measure of the true mean event time and transfer function lag. Inaccuracy in the measure obscures the nature of the relationship.

Another possibility is related to the length of time necessary to find topographic failures: in general, it takes subjects more time to identify a failure topographically than some other way. (For example, the average time to failure diagnosis for topographic failures for Session 2 data is 124.95s; for symptomatic failures, 45.07s; and for serendipitous failures, 60.03s.) Because of the longer times (note in particular the time from first symptom emergence to failure diagnosis for each subject), the complexity measure is not related to the lag values. Failure diagnosis time in this case is more dependent upon the probabilistic nature of the queueing network than the skills or thresholds of individual subjects.

Summary. The preceding discussion indicates that the variables and lags present in the transfer functions are reasonable, if not entirely

explainable. The real time values of the lags frequently agree with the average inter-failure event times calculated from subject data files. A comparison of these values for Session Two data suggests that certain recurring patterns of agreement exist between the lags and inter-event times. These recurring patterns are useful in terms of explaining the presence of both positive and negative terms in the transfer functions. Differences between time values can probably be accounted for by any of several reasons, including the high variability present within the data, the subjective nature of the modelling process, and the existence of events other than failure occurence or symptom emergence (e.g., diagnosis time for a particular system level or subsystem) that affect parameters in the transfer functions.

Results reported in this section demonstrate how two different dimensions of complexity, structural and strategic, can be related to human fault diagnosis skills in a large scale system. The exact nature of the two measures is relatively unimportant beyond a certain degree of intuitive validity. The importance of these results, however, lies in the demonstration that the complexity measures are dependent upon the number of failures in the system and the rate at which their symptoms emerge. These factors are highly dependent upon both system characteristics (i.e., number of levels and degree of redundancy) and subject strategy. Of equal importance is the demonstration that the complexity measures relate to performance in a time-varying manner, and the nature of this time-varying manner is highly dependent upon events that occur within the system and the strategy of individual subjects.

CONCLUSION

The experiment, results, and conclusions in this paper have considered the relationship between the design of a large scale system and human monitoring and control behavior. System characteristics such as number of levels and degree of interconnectedness can have a very strong effect on the ability of humans to maintain proper system operation in the presence of failures. Since normal system operation tends to be affected in the opposite direction in the presence of the same design characteristics, system designers must be careful to create environments that support both system and human performance.

Some rather straightforward measures were used to assess the complexity of a large scale system as it relates to the task of monitoring and control. Complexity, as discussed in this paper, is a dynamic property of a human-machine system. Complexity varies with time and it varies among operators. Furthermore, complexity is multi-dimensional; two dimensions of complexity (i.e., structural and strategic) have been proposed, and it appears that this distinction is useful, both conceptually and practically. Complexity is not due solely to the structure of the system, although a system may certainly be complex due to its structure. Rather, complexity also arises when the human, trying to solve problems within the system's environment, does not understand the structure, and, as a result issues an inappropriate command, misinterprets display information, etc. In short, systems are also complex due to the human's understanding of the system as reflected by his strategy.

Another result from this work concerns the outcome of complexity. Based on a review of the literature and the major control task of subjects (i.e., finding failures), average time to failure diagnosis was

used as the major dependent measure of complexity. As results suggest, however, average time to failure diagnosis alone does not completely describe the implications of complexity. For example, the most complex systems resulted in shorter failure diagnosis times due to the number and location of failures. A smaller fraction of the total number of failures was diagnosed, however. Thus, fraction failures diagnosed was used to explain a different aspect of performance related to task complexity. In short, the result of complexity is multi-dimensional. A single dimension does not capture the outcome of a complex system.

These comments are important in light of the relationships among system characteristics that contribute to complexity, proper operation of the system, and complexity of monitoring and control by the human. As the system becomes more "complex" (from a non-behaviorist's perspective, i.e., more levels and more redundancy), it becomes more resistant to the effects of system failures. Failures take longer to propagate through the more complex systems. Moreover, the effects of any one failure on overall system performance are minimized due to the number of alternate paths through the system. Hence, normal system operation is enhanced. This situation is analogous to the use of redundant or stand-by equipment in systems to increase fault tolerance. On the other hand, as the system becomes more complex, the task of finding system failures becomes more difficult. Although the system design characteristics can help to avoid the short term effects of failures, they can have the dual effect of making the human supervisory controller's task more difficult.

The relationship between complexity and human performance takes on increasing importance given the growing prevalence of large scale systems. Human abilities and limitations in monitoring and controlling

these complex systems must be identified in order to design systems that facilitate good failure diagnosis and network management performance. In short, systems must be designed such that they do not overload human information processing capabilities. Beyond the issue of design, an understanding of human performance constraints should facilitate the creation of effective performance aids. Such aids can be used to help people overcome their limitations in coping with the complex environments these systems create, thereby leading to safe and effective system performance.

ACKNOWLEDGEMENT

This work was supported by the Army Research Institute for the Behavioral and Social Sciences under Contract MDA 903-82-C-0145.

REFERENCES

[1] J. Rasmussen and W.B. Rouse, eds., Human Detection and Diagnosis of System Failures, New York: Plenum Press, 1981.

[2] R.L. Henneman and W.B. Rouse, "Human Performance in Monitoring and Controlling Hierarchical Large-Scale Systems," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-14, no. 2, pp. 184-191, March/April 1984.

[3] S.A. Cook, "An overview of computational complexity," Communications of the ACM, vol. 26, no. 6, pp. 401-408, June 1983.

[4] J. Hartmanis and J.E. Hopcroft, "An overview of the theory of computational complexity," Journal of the ACM, vol. 18, no. 3, pp. 444-475, July 1971.

[5] H.R. Lewis and C.H. Papadimitriou, "The efficiency of algorithms," Scientific American, vol. 238, no. 1, pp. 96-109, January 1978.

[6] R.G. Parker and R.L. Rardin, "An overview of complexity theory in discrete optimizations: Part I. Concepts," IIE Transactions, vol. 14, no. 1, pp. 3-10, 1982a.

[7] R.G. Parker, and R.L. Rardin, "An overview of complexity theory in discrete optimization: Part II. Results and implications," IIE Transactions, vol. 14, no. 2, pp. 83-89, 1982b.

[8] S.R. Das, Z. Chen, T. Lin, and C.L. Sheng, "Complexity and performance of a graph theory algorithm for cluster analysis," International Journal of Computer Mathematics, vol. 13, pp. 41-50, 1983.

[9] B. Borovac, M. Vukobratovic, and D. Stokic, "Analysis of the influence of actuator model complexity on manipulator control synthesis," Mechanism and Machine Theory, vol. 18, no. 2, pp. 113-122, 1983.

[10] M.L. Fredman, "Observations concerning the complexity of a class of on-line algebraic problems," IEEE Transactions on Computers, vol. C-30, no. 1, pp. 83-86, January 1981.

[11] R.E. Ladner, "The complexity of problems in systems of communicating sequential processes," Journal of Computer and System Sciences, vol. 21, pp. 179-194, 1980.

[12] W.B. Rouse and S.H. Rouse, "Measures of complexity of fault diagnosis tasks," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-9, no. 11, pp. 720-727, November 1979.

[13] H. Fujiwara and K. Kinoshita, "On the computational complexity of system diagnosis," IEEE Transactions on Computers, vol. C-27, no. 10, pp. 881-885, October 1978.

[14] R.W. Priester and J.B. Clary, "New measures of testability and test complexity for linear analog failure analysis," IEEE Transactions on Computers, vol. C-30, no. 11, pp. 884-888, November 1981.

[15] S. Henry and D. Kafura, "Software structure metrics based on information flow," IEEE Transactions on Software Engineering, vol. SE-7, no. 3, pp. 510-518, September 1981.

[16] M.H. Halstead, Elements of software science, New York: Elsevier, 1977.

[17] N.S. Coulter, "Software science and cognitive psychology," IEEE Transactions on Software Engineering, vol. SE-9, no. 2, pp. 166-171, March 1983.

[18] V.Y. Shen, S.D. Conte, and H.E. Dunsmore, "Software science revisited: a critical analysis of the theory and its empirical support," IEEE Transactions on Software Engineering, vol. SE-9, no. 2, pp. 155-165, March 1983.

[19] T.J. McCabe, "A complexity measure," IEEE Transactions on Software Engineering, vol. SE-2, no. 4, pp. 308-320, December 1976.

[20] J.S. Davis, "An investigation of chunk based complexity measures," Ph.D. dissertation, Georgia Institute of Technology, 1984.

[21] B.D. Chaudhary and H.V. Sahasrabuddhe, "Two dimensions of program complexity," International Journal of Man-Machine Studies, vol. 18, pp. 505-511, 1983.

[22] W. Weaver, "Science and Complexity," American Scientist, Autumn, 1948.

[23] H.A. Simon, Sciences of the artificial, 2nd edition, The MIT Press: Cambridge, MA, 1982.

[24] W.R. Ashby, An introduction to cybernetics, John Wiley and Sons: New York, 1956.

[25] C. Alexander, Notes on the synthesis of form, Harvard University Press: Cambridge, MA, 1964.

[26] R.D. Brunner and G.D. Brewer, Organized complexity, The Free Press: New York, 1971.

[27] R.J. Waller, "Complexity and the boundaries of human policy making," International Journal of General Systems, vol. 9, pp. 1-11, 1982.

[28] A.J. Mackinnon and A.J. Wearing, "Complexity and decision making," Behavioral Science, vol. 25, pp. 285-296, 1980.

[29] I. Bazovsky, Reliability theory and practice, Prentice-Hall, Inc.: Englewood Cliffs, NJ, 1961.

[30] A.S. Goldman and T.B. Slattery, Maintainability: a major element of system effectiveness, John Wiley and Sons: New York, 1964.

[31] W.H. von Alven, Reliability engineering, Prentice-Hall, Inc.: Englewood Cliffs, NJ, 1964.

[32] W.R. Ashby, Design for a brain, John Wiley and Sons: New York, 1960.

[33] L. Lofgren, "Complexity of descriptions of systems: a foundational study," International Journal of General Systems, vol. 3, pp. 197-214, 1977.

[34] R. Rosen, "Complexity as a system property," International Journal of General Systems, vol. 3, pp. 227-232, 1977.

[35] P.T. Saunders and M.H. Ho, "On the increase in complexity in evolution II. The relativity of complexity and the principle of minimum increase," Journal of Theoretical Biology, vol. 90, pp. 515-530, 1981.

[36] S.N. Greenberg and L.E. Krueger, "Effect of letter orientation and sequential redundancy on the speed of letter search," Memory and Cognition, vol. 11, no. 2, pp. 181-191, 1983.

[37] M.H. Bornstein and M.D. Monroe, "Chromatic information processing: rate depends on stimulus location in the category and psychological complexity," Psychological Research, vol. 42, pp. 213-225, 1980.

[38] G.R. Dirkin, "Cognitive tunneling: use of visual information under stress," Perceptual and Motor Skills, vol. 53, pp. 191-198, 1983.

[39] H.L. Dee and H.J. Hannay, "Reversal of asymmetry in human perceptual performance as a function of labeling, mode of response, and familiarity," Perceptual and Motor Skills, vol. 52, pp. 183-193, 1981.

[40] J. Hochberg and V. Brooks, "The psychophysics of form: reversible-perspective drawings of spatial objects," The American Journal of Psychology, vol. 73, no. 3, p. 337-354, September 1960.

[41] P.C. Vitz and T.C. Todd, "A model of the perception of simple geometric figures," Psychological Review, vol. 78, no. 3, pp. 207-228, 1971.

[42] D.L. Butler, "Predicting the perception of three-dimensional objects from the geometrical information in drawings," Journal of Experimental Psychology, vol. 8, no. 5, pp. 674-692, 1982.

[43] F. Attneave, "Physical determinants of the judged complexity of shapes," Journal of Experimental Psychology, vol. 53, no. 4, pp. 221-227, April 1957.

[44] R. Kimchi and S.E. Palmer, "Form and texture in hierarchically constructed patterns," Journal of Experimental Psychology: Human Perception and Performance, vol. 8, no. 4, pp. 521-535, 1982.

[45] H.A. Simon, "Complexity and the representation of patterned sequences of symbols," Psychological Review, vol. 79, no. 5, pp. 369-382, September 1972.

[46] J.B. Brooke and K.D. Duncan, "Effects of system display format on performance in a fault location task," Ergonomics, vol. 24, no. 3, pp. 175-189, 1981.

[47] J.A. Glover, B.S. Plake, and J.W. Zimmer, "Distinctiveness of encoding and memory for learning tasks," Journal of Educational Psychology, vol. 74, no. 2, pp. 189-198, 1982.

[48] M.A. McDaniel, "Syntactic complexity and elaborative processing," Memory and Cognition, vol. 9, no. 5, pp. 487-495, 1981.

[49] M.H. Ashcraft and E.H. Stazyk, "Mental addition: a test of three verification models," Memory and Cognition, vol. 9, no. 2, pp. 185-196, 1981.

[50] E.F. Loftus and P. Suppes, "Structural variables that determine problem-solving difficulty in computer-assisted instruction," Journal of Educational Psychology, vol. 63, no. 6, pp. 531-542, 1972.

[51] B.B. Morgan and E.A. Alluisi, "Acquisition and performance of a problem-solving skill," Perceptual and Motor Skills, vol. 33, pp. 515-523, 1971.

[52] D.E. Kieras and P.G. Polson, "An approach to the formal analysis of user complexity," International Journal of Man-Machine Studies, in press.

[53] J.G. Wohl, "Maintainability prediction revisited: diagnostic behavior, system complexity, and repair time," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-12, no. 3, pp. 241-250, 1982.

[54] J.G. Wohl, "Cognitive capability vs. equipment complexity in electronic maintenance," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-13, no. 4, 1983a.

[55] J.G. Wohl, J.G., "Connectivity as a measure of problem complexity in failure diagnosis," Proceedings of the Human Factors Society - 27th Annual Meeting, pp. 681-684, 1983b.

[56] R.L. Henneman, "Human problem solving in complex hierarchical large scale systems," Ph.D. thesis, Georgia Institute of Technology, January 1985.

[57] G.E.P. Box and G.M. Jenkins, Time series analysis: forecasting and control, Holden-Day: San Francisco, 1976.

[58] D.C. Montgomery and G. Weatherby, "Modeling and forecasting time series using transfer function and intervention methods," AIIE Transactions, vol. 12, no. 4, pp. 289-306, 1980.

List of Figures

List of Tables

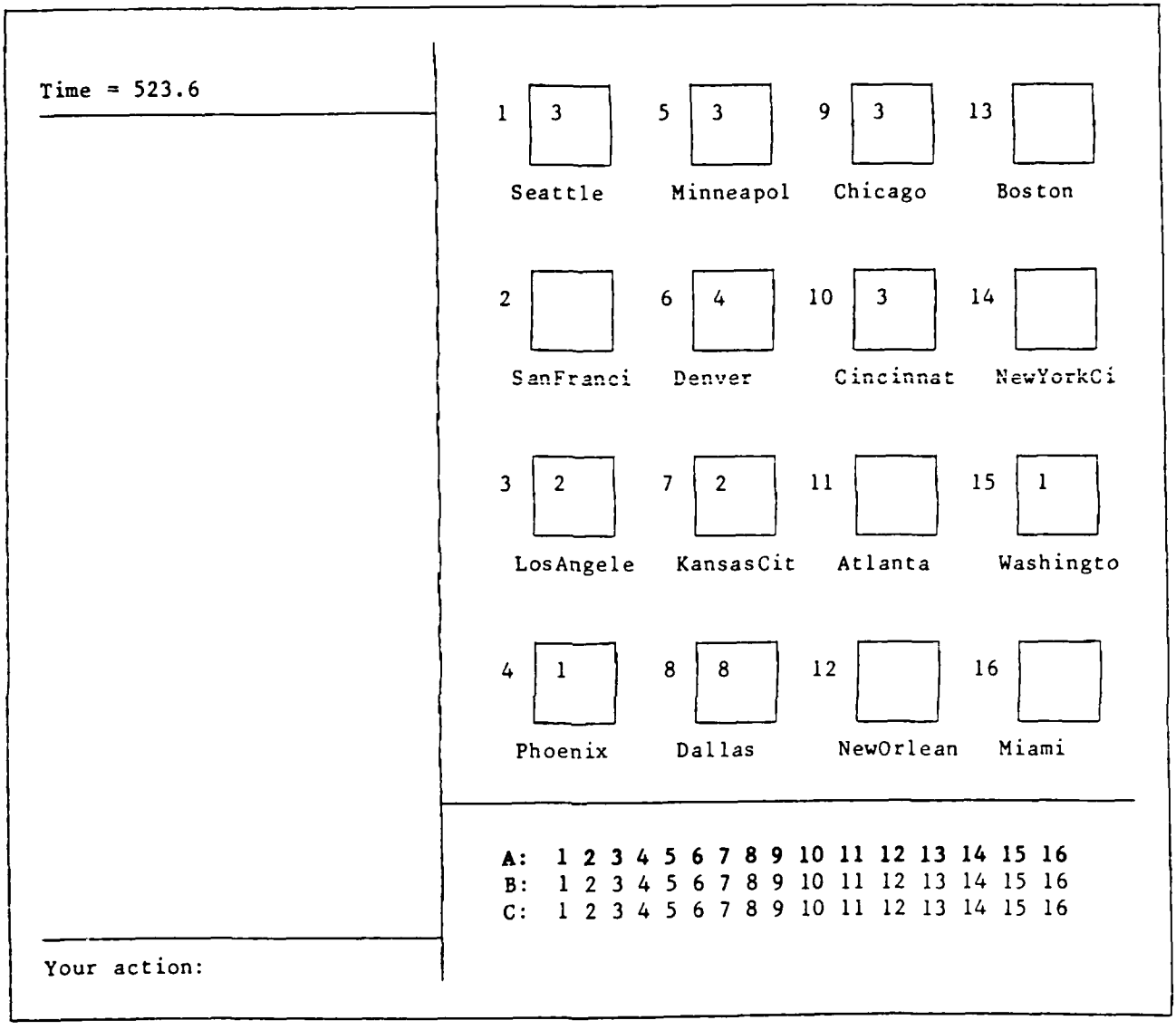


Figure 1 CAIN Display

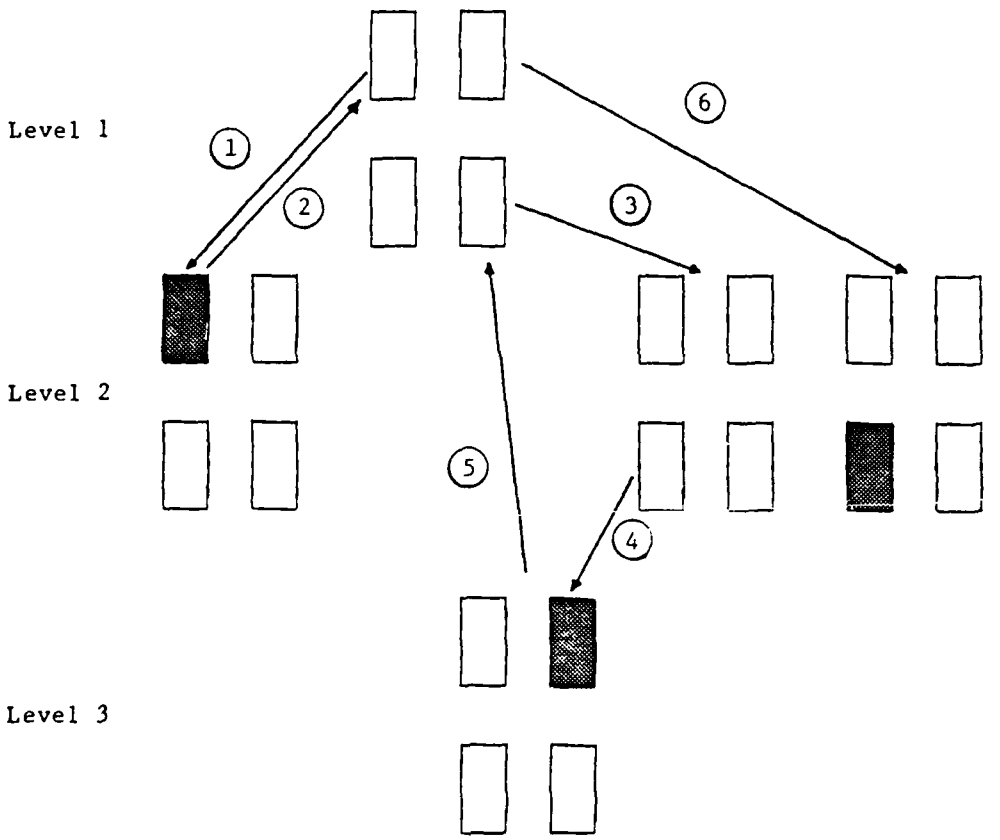


Figure 2 Example calculation of structural complexity

Monitor Display

Cluster	Number of customers
Denver	8
Los Angeles	5
Chicago	1
New York	1

Cluster Display

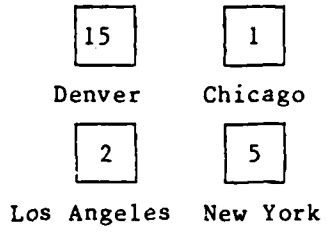


Figure 3 Example monitor and cluster display for calculation of strategic complexity

NO. OF LEVELS	REDUNDANCY: HIGH	REDUNDANCY: LOW
2	LOW Structural Complexity MODERATE Strategic Complexity	MODERATE Structural Complexity LOW Strategic Complexity
3	MODERATE Structural Complexity HIGH Strategic Complexity	HIGH Structural Complexity LOW Strategic Complexity

Figure 4 Summary of results from analysis of variance of complexity measures

	Levels	Fraction Failures Diagnosed	Avg. Time to Failure Diagnosis
Structural C	2	-	
	3	-	-
Strategic C	2	-	
	3	+	

Figure 5 Summary of results from correlation analysis

Table 1 Summary of CAIN commands

ACCESS Commands

dCITY	down CITY
u	up one level
fCITY	find CITY
a	return to top level

MONITOR Commands

m	monitor
s	system statistics
w	watch list
+CITY	add CITY to watch list
-CITY	delete CITY from watch list
o	list repair orders
e	erase warning message from bottom of screen

DIAGNOSTIC Commands

t	tests displayed cluster of nodes
cCITY	information about CITY

CONTROL Commands

rCITY	replaces equipment in CITY
lCITY=%load	alters load in cluster CITY to %load
lsys=%load	alters load in entire system
l	displays load

Table 2 Example calculation of strategic complexity

Cluster	U	p[F|X]		U x p[F|X]
	uncertainty	monitor	cluster	
Denver	20.12	.600	.750	15.090
Los Angeles	0.54	.100	.015	.054
Chicago	7.36	.001	.001	.007
New York	9.12	.001	.050	.456
			Strategic Complexity =	15.607

Table 3 Summary of average times from failure-related events to failure diagnosis (Session 2)

	All	Topographic	Symptomatic	Serendipitous
Subject 1				
Total Failures	58	20	15	23
Frac. of Total		0.35	0.26	0.40
$\bar{T}$(Failure)	63.02 + struct 48.1	81.68	27.47	69.99 + struct 48.1
$\bar{T}$(Symptom)	29.65	38.18	14.98	31.81
Subject 2				
Total Failures	61	10	18	33
Frac. of Total		0.16	0.30	0.54
$\bar{T}$(Failure)	75.85 + strat 71.1	101.03	47.66	83.59
$\bar{T}$(Symptom)	32.29 - struct 22.4	42.99	24.44 - struct 22.4	33.33 - struct 22.4
Subject 3				
Total Failures	66	16	17	33
Frac. of Total		0.24	0.26	0.50
$\bar{T}$(Failure)	78.04 + struct 60.1	180.32	42.52 - struct 40.1	46.75 - struct 40.1
$\bar{T}$(Symptom)	44.29 - struct 40.1	97.88	19.57 + struct 10 - strat 16.7	31.03 - struct 23.4
Subject 4				
Total Failures	63	22	19	22
Frac. of Total		0.35	0.30	0.35
$\bar{T}$(Failure)	53.90 + strat 46.4	78.23	34.06 + strat 23.2	46.71 + strat 46.4
$\bar{T}$(Symptom)	31.80 - struct 11.6	47.56	14.45 - struct 11.6	31.03 - struct 11.6

Subject 5				
Total Failures	75	16	23	36
Frac. of Total		0.21	0.31	0.48
$\bar{T}$(Failure)	66.93 + strat 67.6	146.20	50.90	41.93
$\bar{T}$(Symptom)	28.47	56.91	19.98 + strat 11.6	21.25 + strat 11.6
Subject 6				
Total Failures	43	16	20	7
Frac. of Total		0.37	0.47	0.16
$\bar{T}$(Failure)	64.30 + struct 66.2	94.48	39.31	66.75
$\bar{T}$(Symptom)	31.01 - struct 29.4	52.16	10.44	41.43
Subject 7				
Total Failures	97	17	15	65
Frac. of Total		0.18	0.16	0.67
$\bar{T}$(Failure)	78.89 + struct 49.8 + strat 49.8	143.59	57.35 + struct 49.8 + strat 49.8	66.94
$\bar{T}$(Symptom)	45.74	88.15	26.42	39.11
Subject 8				
Total Failures	47	12	8	27
Frac. of Total		0.26	0.17	0.57
$\bar{T}$(Failure)	103.95	174.07	61.32	85.42
$\bar{T}$(Symptom)	65.01 - struct 44.2	116.95	24.69	53.88 - struct 44.2

Table 4 Summary of average event times and lag values

Average time	lag value	Average time	lag value
63.0	48.1	19.6	10.0
75.9	71.1	19.6	16.7
32.3	22.4	34.1	23.2
78.0	60.1	14.5	11.6
44.3	40.1	20.0	11.6
53.9	46.4	57.4	49.8
31.8	11.6	70.0	48.1
66.9	67.6	33.3	22.4
64.3	66.2	46.8	40.1
31.0	29.4	31.0	23.4
78.9	49.8	46.7	46.4
65.0	44.2	31.0	11.6
24.4	22.4	21.3	11.6
42.5	40.1	53.9	44.2